DRAFT

Authorization Server

Protection Profile (ASPP)

For

Basic Robustness Environments

Version 0.41
05 April 2004

Prepared for
National Security Agency
9800 Savage Road
Fort Meade MD, 20755

Prepared by
Science Applications International Corporation
7125 Gateway Drive, Suite 300
Columbia, MD 21046

Foreword

This publication, Authorization Server Protection Profile for Basic Robustness Environments, is issued by the National Security Agency as part of its program to promulgate security standards for information systems.

Comments on this document should be directed to Troy Young, National Security Agency, V51, 9800 Savage Road, Ft. Meade, MD 20755.

Version 0.4 18 November 2003

Protection Profile Title:

Authorization Server Protection Profile for Basic Robustness Environments.

Criteria Version:

This Protection Profile (PP) was developed using Version 2.1 of the Common Criteria (CC) [1] and applying the NIAP interpretations that have been approved by TTAP/CCEVS Management as of July 10, 2002.

Constraints:

Targets of Evaluation (TOEs) developed to satisfy this Protection Profile shall conform to CC Part 2 and CC Part 3 and applicable NIAP approved interpretations.

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

This section contains document management and overview information necessary to allow a Protection Profile (PP) to be registered through a Protection Profile Registry. The Identification provides the labeling and descriptive information necessary to identify, catalogue, register, and cross-reference a PP. The Overview summarizes the profile in narrative form and provides sufficient information for a potential user to determine whether the PP is of interest. The Overview can also be used as a stand-alone abstract for PP catalogues and registers. The Conventions section provides an explanation the Common Criteria (CC) notation and formatting plus outlines how this document is organized. The Terms section gives a basic definition of terms, which are specific to this PP. Finally, the Related Profiles section identifies profiles directly related to this profile and may be of interest to those interested in this profile.

1.1 Protection Profile Identification

Title: Authorization Server Protection Profile (ASPP) for Basic Robustness Environments

Sponsor: National Security Agency (NSA)

CC Version: Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.1, August 1999, ISO/IEC 15408-2. Part 2 extended Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.1, August 1999, ISO/IEC 15408-3. Part 3 conformant Evaluation Assurance Level 2 (EAL 2) augmented with ALC_FLR.2, and AVA_MSU.

Registration: <to be provided upon registration>

Protection Profile Version: Version 0.1, dated April 30, 2003

Keywords: authorization, access control, Enterprise Access Management (EAM), Privilege Management Infrastructure (PMI), Authorization Service

1.2 Protection Profile Overview

The CC ASPP specifies a set of functional security and assurance requirements for Information Technology (IT) products. The Authorization Server is a family of software products that supports access control of IT resources (e.g., web servers, databases, application servers, individual web pages, and specific data files/objects). Authorization Server software provides a capability to map a user's identity to a set of privilege attributes. It also provides a mechanism to assign access requirements to IT resources. The authorization service then executes pre-defined rules or policies which compares a user's privilege attributes to the requested IT resources access requirements to make an access control decision. The access control decision is generally enforced via an Authorization Server component on a web server typically called an "agent."

Authorization Server software can also provide an Application Programming Interface (API) that enables designated custom application to obtain access control decisions from the Authorization Server's policy engine. A second API is generally available to enable custom applications and

databases to obtain authorized user attributes to enable them to make their own access control decisions. In this case the Authorization Server is serving as an "Attribute Authority."

The deployment of commercial Authorization Server software can also be characterized as a "Privilege Management Infrastructure" (PMI). The PMI can be defined as the processes and software required to operate an "Authorization Service."

ASPP-conformant products provide access control services to web and applications servers, plus provide APIs to provide authorizations decisions and attributes to custom applications. ASPP-conformant products also provide the ability to protect themselves and their associated data from unauthorized access or modification while ensuring accountability for authorized actions.

The ASPP is a "software only" PP dependent on the IT environment (hardware, operating system, and other software products) to meet all the security functional and assurance requirements for a Basic Robustness environment (as defined by the NSA Information Assurance Directorate (IAD) document "Protection Profile (PP) Consistency Guidance for Basic Robustness"). This ASPP provides a level of protection that is appropriate for IT environments that have main Authorization Server components on a private protected network (e.g., behind firewalls) and administered by highly trusted users. The ASPP does not fully address threats posed by malicious administrative or system development personnel. ASPP-conformant products are suitable for use in both commercial and government environments.

The ASPP was constructed to provide a target and metric for the development of Authorization Server software. This PP identifies security functions and assurances representative of the lowest common set of requirements that should be addressed by a useful Authorization Service. Targets of Evaluation (TOEs) compliant with this PP will meet the assurance requirements of Evaluation Assurance Level (EAL) 2 augmented.

This PP defines:

- Assumptions about security aspects of the environment in which the TOE will be used;
- Threats that are to be addressed by the TOE;
- Organizational security policies pertaining to the TOE;
- Security objectives of the TOE and its environment;
- Functional and assurance requirements to meet those security objectives; and
- Rationale demonstrating how the requirements meet the security objectives, and how the security objectives address the threats.

1.3 Conventions

The notation, formatting, and conventions used in this PP are largely consistent with those used in version 2.1 of the CC. Selected presentation choices are discussed here to aid the PP user.

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph 2.1.4 of Part 2 of the CC. Each of these operations is used in this PP.

> The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**.
>
> The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by *italicized text*.
>
> The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignment is indicated by showing the value in square brackets, [Assignment value].
>
> The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration number).
>
> The **Security Target (ST) author** operation is used to denote points in which the final determination of attributes is left to the ST writer. ST writer operations are indicated by the words "determined by the ST Author."

The CC paradigm also allows PP and ST authors to create their own requirements. Such requirements are termed 'explicit requirements' and are permitted if the CC does not offer suitable requirements to meet the authors' needs. Explicit requirements must be identified and are required to use the CC class/family/component model in articulating the requirements. In this PP, explicit requirements will be indicated with the "EXP" appended to the family name.

Application Notes are provided to help the developer, to either clarify the intent of a requirement, identify implementation choices, or define "pass-fail" criteria for a requirement. For those components where Application Notes are appropriate, the Application Notes will follow the requirement component.

1.4 Related Protection Profiles

There are no PPs that directly relate to the Authorization Server software. However, the following PPs provide security requirements to components that make up the IT Environment in which the Authorization Server software is deployed:

Web: Web Server Protection Profile, Web Browser Protection Profile Draft, Version: .6, dated 31 July 2001

Operating Systems: Controlled Access (Basic Robustness/C2) (CAPP) Version. 1.d, dated 8 October 1988

Public Key Infrastructure (PKI): Certificate Issuing and Management Components Family of Protection Profiles, Version 1.0, dated 31 October 2001.

1.5 Protection Profile Organization

Section 1, PP Introduction, provides the document management and overview information necessary to identify the PP along with references to other related PPs.

Section 2, TOE Description, defines the TOE and establishes the context of the TOE by referencing generalized security requirements.

Section 3, TOE Security Environment (TSE), describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment.

Section 5, IT Security Requirements, defines the security functional and assurance requirements derived from the CC, Part 2 and Part 3, respectively, that must be satisfied by the TOE, the TOE IT environment, and the Non-IT environment.

Section 6, Rationale, provides rationale to demonstrate that the security objectives satisfy the threats and policies. This section also explains how the set of requirements are complete relative to the security objectives. This section includes a dependency analysis, Strength of Function (SOF) discussion, and rational for the use of explicit requirements.

Section 7, References, provides background material for further investigation by users of the PP.

Section 8, Terminology, provides a listing of definitions of terms.

Section 9, Acronyms, provides a listing of acronyms used throughout the document.

2 TOE DESCRIPTION

This PP specifies the minimum security requirements for a TOE composed of several "software only" components, which together, make up an Authorization Server system. The purpose of an Authorization Server is to provide an organization with a web access management solution that helps to enable secure access to web-based resources. These commercial security products enhance website security management by providing a platform for centrally managing access to all web resources and applications. In a large organization, this is cost saving over building proprietary user directories and access control systems into individual applications. The authorization policy management feature of these products enables central or distributed management of user access privileges. The products also provide for the creation of business or policy rules, often called rulesets, which can incorporate both static (such as a role) or dynamic user attributes (such as a user's checking account balance) to define the access control requirements to protect web-based resources (e.g.: Universal Resource Locators (URLs), files, and objects). Also provided is a web Single Sign-on (SSO) capability that allows users to navigate across web-based resources, both within a single site and across multiple sites, while authenticating only once. Most Authorization Server products also provide a software component, often called a web or application server "agent," which provides the access control decision enforcement point for web and application servers.

In addition to web and application server access management with agents, most Authorization Server software packages provide an API to enable designated custom applications to obtain access control decisions from the Authorization Server's policy engine. A second API is generally available to enable custom applications and databases to make their own access control decisions by obtaining authorized user attributes. In this last mode of operation, the Authorization Server software functions as a privileged "Attribute Authority" interface.

The following describes the software components that generally make up an Authorization Server product. The components are listed in three subsections depicting different operational scenarios: web/application servers, API for access control decisions, and API for Attribute Authority.

2.1 TOE for Web Server Access Control

The following Authorization Server and IT Environment software components are required for an authorization service that provides access control to static web resources (e.g.: URLs, files, and objects) or web based Java 2 Platform, Enterprise Edition (J2EE) application servers program (e.g., URLs, or Enterprise Java Beans (EJB)). All the software components will be described in order to fully understand the access control scenario for web-based resources.

2.1.1 Authorization Server Administrative Interface

This software component, or sometimes called a "server," provides the main interface for administrative users to manage the system. This component, which would reside in a protected enclave, would provide the administrative user interface, a user attribute management capability, and an access policy management tool.

2.1.1.1 Administrative User Interface

This provides the server capabilities required to allow remote users to securely log on and gain access to the product's management tools. Users gain access to this component either via a web based interface or a client/server interface, depending on the products design.

2.1.1.2 User Attribute Management

This component provides the tools to create and modify user privileged attributes or entitlements. This includes creating "groups" or "communities of interest" as well as adding members to those groups. It also includes changing values for existing multi-valued attributes in the User Attribute Data Store (UADS). This component requires a secure interface to the UADS to make these management changes. NOTE: The UADS is part of the TOE's IT environment required for an authorization server to function, however, it is NOT part of the TOE. Further details are provided in the TOE environment section of this PP (section 3).

2.1.1.3 Access Policy Management

This software component provides the tools to define IT resources to be protected and define the required access criteria. Each policy or ruleset defines who can access each resource, the conditions under which access will be allowed, and the user entitlement or privilege attribute information needed for a successful authorization. The software should enable an organization to build Web access and user privilege policies based on dynamic user profile data. Business or access control rules should be translated into an online access management policy using native language and Boolean constructs to provide access based on business requirements or government policy. This component requires a secure interface to the Resource Access Requirements or Policy Data Store to make these management changes. NOTE: The Resource Access Requirements or Policy Data Store is part of the TOE's IT Environment required for an authorization server to function, however, it is NOT part of the TOE. Further details are provided in the TOE Environment section of this PP (section 3).

2.1.2 Authorization Server Policy Decision Engine

This software component, sometimes called a "server," provides a mapping between the required access criteria (policy/ruleset) for a web based resource and user privilege attributes or entitlements. It performs the required computation to make an access control decision. The policy's can be "Boolean" constructs (with AND, OR, NOT and grouping operators) or , policies represented with multi-valued attributes, relative comparisons (less than, greater than) and other more complex situations. Although most policies are deterministic (always gives the same answer for the same inputs) a policy could be based on input from a random number generator. This component, which would reside in a protected enclave, would require secure interfaces to the agent and to the data stores to obtain the information needed to make the policy decision.

2.1.2.1 Agent Interface

This provides a secure mechanism to obtain the user identification information from the agent. Most authorization server products support multiple forms of user Identification and Authentication (I&A). The most common are user name/passwords and X.509 PKI certificates.

In the case of user name/password I&A, the agent interface software would challenge a user to provide user name and password. The Agent would collect the user name and password and forward it back to the TOE via this Agent interface. The authentication of the user name and password is accomplished by the authorization decision engine which verifies the password submitted was correct by obtaining the user's stored password in the UADS. When certificate-based, client-authenticated TLS is used, I&A is performed during the TLS challenge-response negotiation, whereas use of certificate attributes for making access control decisions constitutes authorization. As part of the TLS setup, the web server can be given the responsibility to validate the certificate's for currency (i.e: date not expired) and trust path for the Certificate Authority. The agent would obtain the Distinguished Name (DN) or other certificate attributes of the user from the certificate and pass that via the Agent Interface for specific user authentication In both user name /password and certificate based I&A the agent would also identify the specific resource (e.g., URL or file) which the user desired access and pass that back to the authorization engine via this interface. Finally, this interface would be used to pass back the access control decision (grant or deny) to the policy enforcement agent. It is recognized that the exact implementation of the I&A may differ between products. An implementation could conduct the initial user I&A solely by the webserver, outside the TOE. Then later the TOE verifies that the authenticated user name is authorized for access to the protected resource.

2.1.2.2 UADS Interfaces

This software component provides a secure mechanism to obtain users' entitlements from the UADS. When a user clicks on a protected web server link, thereby requesting access to a protected web resource, the software will request the user to authenticate by providing his/her identity credentials (usually via user name and password or PKI certificates). The UADS interface enables the Policy Decision engine to authenticate the user by verifying the claimed identity matches the user's password or DN stored in the Data Store. This interface is also used to obtain the user entitlement attributes, which are generally "cached" in the Policy Engine to be used for access control decision-making.

2.1.2.3 Policy Data Store Interface

This provides a secure mechanism to communicate with and obtain IT resource access control requirements from the Policy Data Store. These policies can also be "cached" in the Policy Engine to be used for access control decision-making.

2.1.3 Authorization Server Policy Enforcement Agent

This software component, which is generally provided by the authorization server vendor, is designed to be installed on the on the web server or application server which houses the resources need to be protected. These agents generally conform to the Web servers' native architecture. For example, there is a *module* for Apache®; a *filter* for Microsoft™ Internet Information Server® (IIS); an *extension* for iPlanet®, and so on. These will be referred to simply as Agents or Web Server Agents throughout this document. NOTE: the web or application server software itself is generally not part of the TOE and neither is part of the evaluation. Essentially, these Agents replace or augment the Web server's native security mechanisms. The Agent runs in the same process as the Web server itself and is invoked

whenever the Web server needs to determine access rights for a particular Uniform Resource Identifier (URI). The Web Server Agent forwards access requests and users identify information to an Agent Interface component on the Authorization Server Policy Engine. Agent Interface component, working with the other component, will authenticate the user based on verifying the user name / password or DN matches the values stored in the UADS. The Policy Engine make the access control decision and passes the answers back to the Agent (via the Agent Interface component). The Agent is then responsible for passing the answer it receives back to the Web server to enforce the decision by granting or denying the user access to the resource.

2.1.4 Authorization Server Administrative Interface Program

This software component allows administrative users access to the Authorization Server Administrative Interface. This software provides the client capabilities required for remote users to securely log on and gain authenticated access to the product's management tools. This component could be a traditional custom developed "fat" client provided by the Authorization Server vendor to access their Administrative Server. In that case, the client software would be part of the TOE. Some products could allow users to access the Administrative Server via a web-based interface, using either SERVLET or APPLET technology. In the case where the client is a commercial web browser vice a custom developed application, the web browser is NOT part of the ASPP TOE. The browser should be required to meet the security requirements outlined in the "Web Browser Protection Profile."

2.1.5 Authorization Server Decision API

Authorization Server software packages generally provide an API that allows a designated custom application to obtain access control decisions from the Authorization Server's policy engine. This component provides the capability for an alternate interface on the Authorization Server Policy Decision Engine for applications that cannot employ the "agents." Using the administrative tools, policies or rules are written which can be executed by software queries to the Decision API. When the API receives the request for an authorization decision, it must first validate the identity of the requesting software entity and ensure it is authorized to use the API. The API would probably employ the same interfaces to the user attribute and policy data stores as the agent to obtain the access control requirements of the resource being protected and the user's entitlements. The Policy Engine would then make the decision and the API would return a "Grant or Deny" command to the requesting software application.

2.1.6 Authorization Server Attribute Authority API

Authorization Server software packages generally provide an API that enables designated custom applications or databases to obtain user entitlements from the UADS. This API allows the Authorization Server software to function as an organization's central "Attribute Authority" to support various IT resources that need user attributes to make their own access control decisions. When the API receives the request for a user attribute, it must first validate the identity of the requesting software entity and ensure it is authorized to use the API. The API would have an interface to the UADS from which it would obtain the user entitlement. The API would then return the attribute values requested to the application or database making the request.

2.2 Authorization Server Scenarios

The following outlines three basic scenarios in which Authorization Server products provide: (1) access control decision and enforcement services; (2) provide only access control decision support, and (3) provide authorized user attributes to other applications.

2.2.1 Web Server Access Control Scenario

Figure 1 outlines the scenario for an Authorization Server to provide access control over resources on a web or applications server. It also graphically depicts the ASPP TOE software components.

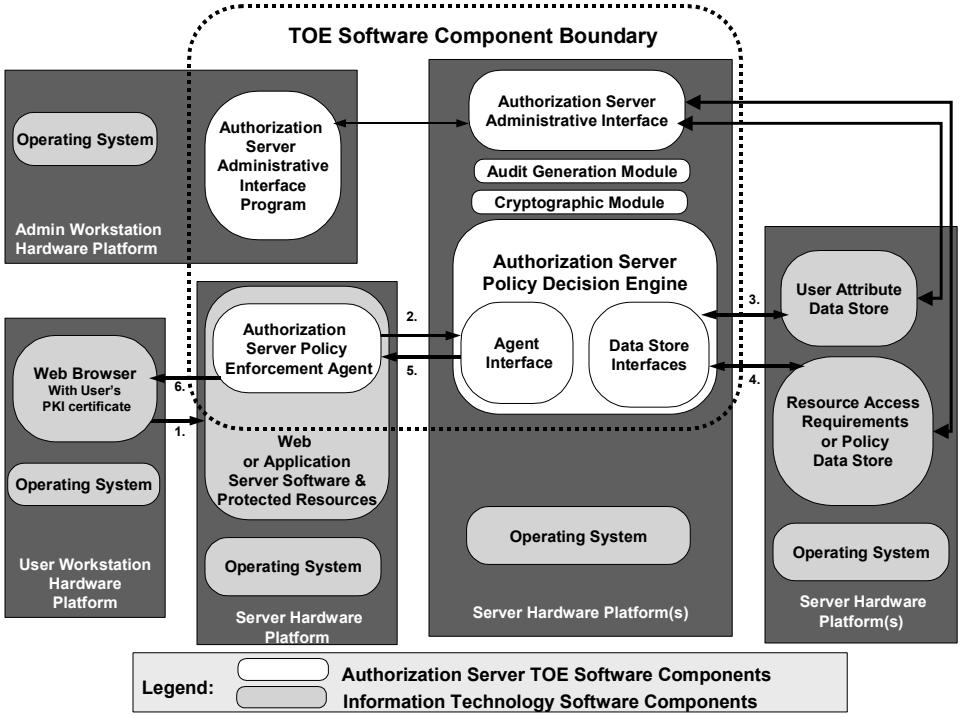

Figure 1 – Web or Application Server Access Control Scenario

The following describes the information between an authorized human user requesting access to a protected web resource and the Authorization Server components depicted above. These steps assume an administrator has previously established access requirements in the Policy Data Store and that the user's entitlements are held in the User Attribute store. The scenario is described using software Public Key Infrastructure (PKI) certificates for I&A.

- Step 1: A user with PKI certificates in his browser, points and clicks on a URL, which is linked to on a web server or application server resource that the user desires to gain access.

- Step 2: The Authorization Server Agent intercepts the access request and the user's DN or other designated attributes from his certificate, and then forwards them to the Policy Engine. As part of this process the Agent must first establish a secure, mutually authenticated path between itself and the agent interface on the Authorization Server Policy Engine.

- Step 3: The Policy Engine first authenticates the user's identity by comparing the DN presented by the user to a valid DN obtained in the UADS. The engine also obtains the user's entitlements and stores them in cache.

- Step 4: The Policy Engine then obtains the access requirements for the request URL from the Policy Data Store.

- Step 5: The Policy Engine compares the Boolean constructs of the access requirements and the user's privilege attribute entitlements and makes an access control decision. The decision to grant or deny access is passed over the network to the Policy Enforcement Agent.

- Step 6: Working with the web or application server, the Agent either allows the user to access the web resource or directs the web server to deny access to the resource (displaying the default "Forbidden" page or a custom developed message).

2.2.2 Authorization Decision API Scenario

The following illustrates the capability of an Authorization Server API that allows designated custom applications to obtain access control decisions from the Authorization Server's policy engine. Figure 2 outlines the scenario to provide authorization decisions to custom SERVLETS running on an applications server or other custom applications. It also graphically depicts the ASPP TOE software components for this scenario.

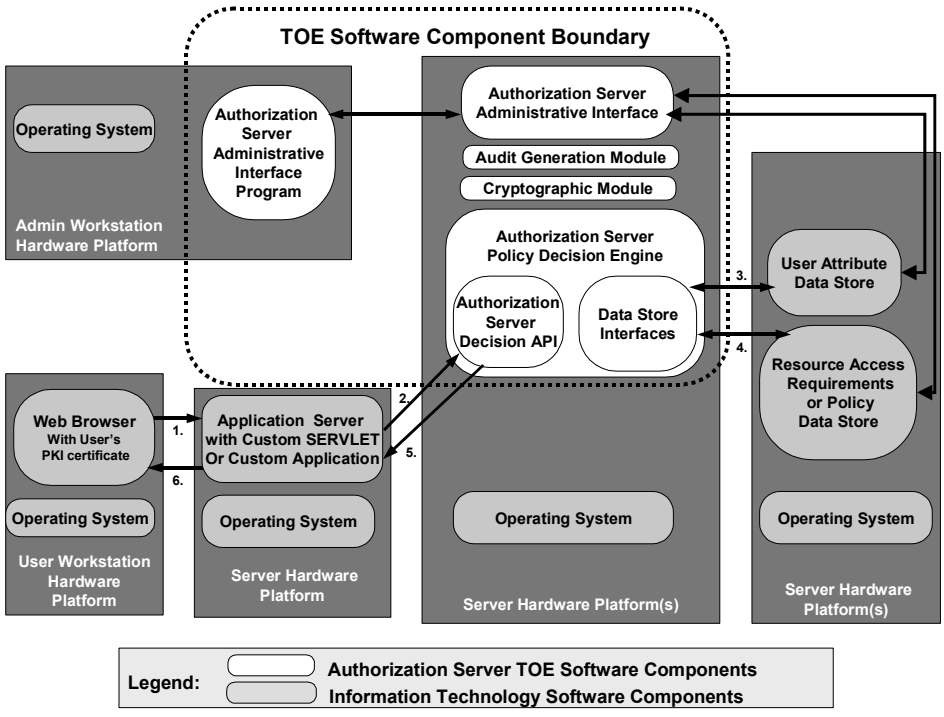

Figure 2 – Authorization Decision API Scenario

The following describes the information between an authorized human user requesting access to a custom application or SERVLET resource and the Authorization Server components depicted

above. These steps assume an administrator has previously established access requirements for the application in the Policy Data Store and that the user's entitlements are held in the User Attribute store. The scenario is described using software PKI certificates for I&A.

- Step 1: A user with PKI certificates in his browser, points and clicks on a URL which is linked to a SERVLET or custom application on an application server (the scenario could also include attempting to execute a specific function within the application).

- Step 2: The SERVLET or custom application is programmed to recognize this resource is protected, and therefore knows it must ask for the authorization before connecting the user. The application must first establish a secure, mutually authenticated path between itself and the Decision API. It then generates the appropriate software commands and sends an access request, along with the user's DN from his certificate, the Policy Engine API.

- Step 3: The Policy Engine first authenticates the user's identity by comparing the DN presented by the user to a valid DN obtained in the UADS. The engine also obtains the user's entitlements and stores them in cache.

- Step 4: The Policy Engine then obtains the access requirements for the application (or function within the application) from the Policy Data Store.

- Step 5: The Policy Engine compares the Boolean constructs of the access requirements and the user's privilege attribute entitlements and makes an access control decision. The decision to grant or deny access is passed over the network to the application that made the request.

- Step 6: Based on the decision, the application's internal code then either allows the user to access the application (or function) or directs the deny access to the application. Depending on the application, the response sent back to the user could be a dynamically built web page allowing the user to see only objects they are authorized to view.

2.2.3 Attribute Authority API Scenario

The following illustrates the capability of an Authorization Server to provide an API to allow designated custom applications and database to obtain user's attributes from the UADS. Figure 3 outlines the scenario to provide user attributes to custom applications or databases. It also graphically depicts the ASPP TOE software components for this scenario.

- Step 1: A user with PKI certificates in his browser, points and clicks on a URL that is linked to a custom application or database. He/she makes a request of the application or database for information.

- Step 2: The application or database is programmed to recognize this resource is protected, and therefore knows it must ask the for the user's attributes before making an access control decision about the data. The application must first establish a secure, mutually authenticated path between itself and the Attribute API. It then generates the appropriate software commands and sends an attribute request, along with the user's certificate DN, the Attribute API.

- Step 3: The API first authenticates the user's identity by comparing the DN presented by the user to a valid DN obtained in the UDAS. The API also obtains the user's entitlements from the UADS.

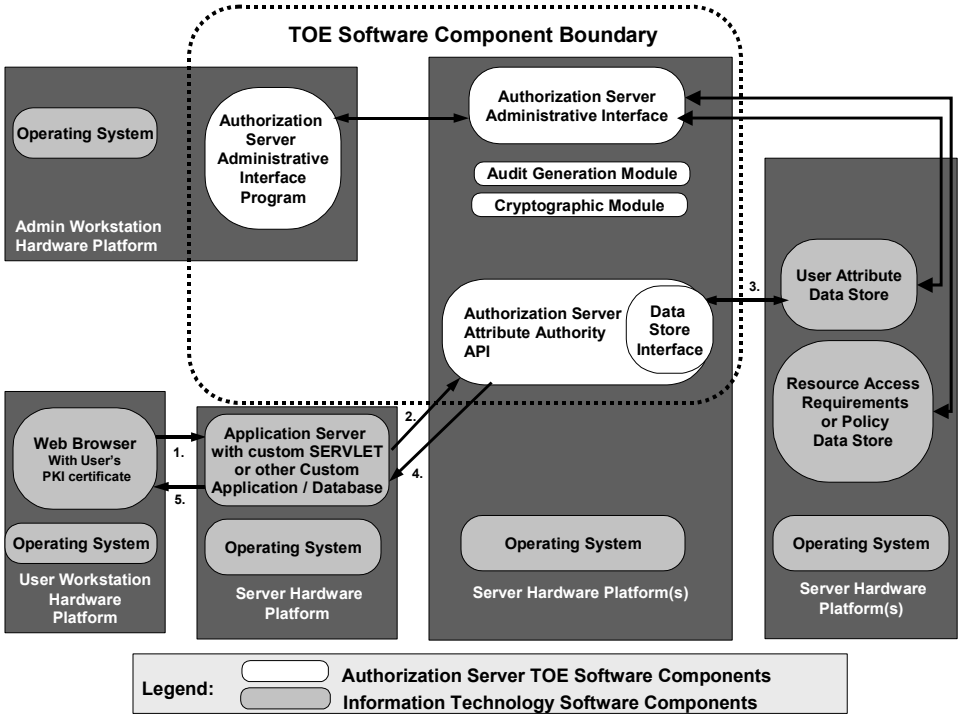

Figure 3 - Attribute Authority API Scenario

- Step 4: The API provides the requested user's attributes to the requesting application or database.

- Step 5: The application or database then makes an access control decision based on the user's attributes. Depending on the application or database, the response sent back to the user could be a dynamically built web page allowing the user to see only objects they are authorized to view.

2.3 Security Features

The TOE security features and functional requirements can be categorized as follows: Identification and Authentication, Administration, Information Flow Control, Encryption, and Audit.

2.3.1 Identification and Authentication

The TOE requires multiple Identification and Authentication (I&A) mechanisms for users desiring to obtain access to web resources protected by the TOE software and for administrative users to access management services residing on the TOE. The type of authentication

mechanism required is dependent on the type of service being requested. An encrypted channel must be established between the TOE and authorized users to protect the transfer of authentication data.

2.3.2 Administration

"Administrators" refers to the roles assigned to the individuals responsible for the installation, configuration, and maintenance of the TOE. The TOE requires two separate administrative roles: Audit Administrator and Security Administrator. The Audit Administrator is responsible for the regular review of the TOE's audit data. The Security Administrator is responsible for all other administrative tasks (e.g., creating the required access security policies (rulesets), user attribute management). The Security Administrator is also responsible for the configuration and maintenance of cryptographic elements related to the establishment of secure connections to and from the TOE. It is important to note that while this PP requires the two administrative roles outlined above, it provides the ST author the option of including additional administrative roles as well. For example, most authorization server products allow for subordinate Security Administrators that can be given limited authority to manage access to specific web resources. This allows for a distributed administration of web resources by local webmasters.

2.3.3 Access Control

Section 5.1.3 defines a minimum set of access control functional requirements that must be met by the TOE. Unlike some PPs, where all the access control functionality is designed to provide requirements to protect just the TOE data, this PP specifies the access control functional requirements for the TOE to provide authorization and access control services over protected web based resources that are not actually part of the TOE. This PP includes the introduction of an "Authorization Server Access Control policy." This is not a single "standard" policy, like Discretionary Access Control (DAC), but rather dynamic policy that is based on the Security Administrator's defined rules or operations. In this concept there are "subjects" which act on behalf of users to gain access to "named objects," where all the operations between subject and object covered by the Authorization Server Access Control policy. The "subjects" are generally the Authorization Server "Agents." The "named objects" are the designated web based resources (web server, directories, files, or objects) that the Authorization Server is protecting. The access control is based on the Security Attributes of both the user and the object. These attributes can be user identity and group membership(s) associated with a subject, the time of day attribute associated with the operation; and access control attributes associated with an object**.**

2.3.4 Encryption

Section 5.1.2 "Cryptographic Support" defines the minimum set of cryptographic attributes required by the TOE. The TOE's cryptographic module must comply with the FIPS PUB 140-2 Level 1 standard and follow with commercially accepted best practices.. Since this PP is primarily a "software only" PP, it is intended for the ST author must implement the "cryptographic module" in software. In that case, the TOE must generate and distribute symmetric and asymmetric keys. However, if a vendor elected to use a hardware cryptomodule,

those modules and products would need to be included in the TOE and be subject to the TFS cryptography requirements. The ST author is provided several implementation selections for key generation and may distribute keys manually, electronically, or both. .

2.3.5 Audit

Section 5.1.1 "Security Audit" describes the TOE's generation of auditable events. Since the TOE will be running on top of an operating system that is compliant with the CAPP the storage, protection, and analysis of the audit records will be a function of the IT environment. The IT environment and the TOE together will cover the alarming aspects as a result of an audit analysis and the overall audit management. Table 3 in the FAU_GEN.1-NIAP-0410 requirement lists the minimum set of auditable events that must be available to the Security Administrator for configuration on the TOE. Each auditable event must generate an audit record. Table 3 also provides a minimum list of attributes that must be included in each audit record. The ST author may include additional auditable events and audit record attributes. If the ST author includes any additional functional requirements not specified by this PP, they must consider any security relevant events associated with those requirements and include them in the TOE's list of auditable events and records.

3 TOE SECURITY ENVIRONMENT

In trying to specify the environments in which TOEs with various levels of robustness are appropriate, it is useful to first discuss the two defining factors that characterize that environment: ***value of the resources*** and ***authorization of the entities*** to those resources.

In general terms, the environment for a TOE can be characterized by the authorization (or lack of authorization) the least trustworthy entity has with respect to the highest value of TOE resources (i.e., the TOE itself and all of the data processed by the TOE).

Note that there are an infinite number of combinations of entity authorization and value of resources; this conceptually "makes sense" because there are an infinite number of potential environments, depending on how the resources are valued by the organization, and the variety of authorizations the organization defines for the associated entities. In Section 3.3, these two environmental factors will be related to the robustness required for selection of an appropriate TOE.

3.1 Value of Resources

Value of the resources associated with the TOE includes the data being processed or used by the TOE, as well as the TOE itself (for example, a real-time control processor). "Value" is assigned by the organization. For example, in the Department of Defense (DoD) low-value data might be equivalent to data marked "For Official Use Only (FOUO)," while high-value data may be those classified Top Secret. In a commercial enterprise, low-value data might be the internal organizational structure as captured in the corporate on-line phone book, while high-value data might be corporate research results for the next generation product. Note that when considering the value of the data one must also consider the value of data or resources that are accessible through exploitation of the TOE. For example, a firewall may have "low value" data itself, but it might protect an enclave with high value data. If the firewall was being depended upon to protect the high value data, then it must be treated as a high-value-data TOE.

3.2 Authorization of Entities

Authorization that entities (users, administrators, other IT systems) have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) is an abstract concept reflecting a combination of the trustworthiness of an entity and the access and privileges granted to that entity with respect to the resources of the TOE. For instance, entities that have total authorization to all data on the TOE are at one end of this spectrum; these entities may have privileges that allow them to read, write, and modify anything on the TOE, including all TSF data. Entities at the other end of the spectrum are those that are authorized to few or no TOE resources. For example, in the case of a router non-administrative entities may have their packets routed by the TOE, but that is the extent of their authorization to the TOE's resources. In the case of an Operating System (OS), an entity may not be allowed to log on to the TOE at all (that is, they are not valid users listed in the OS's user database)**.**

It is important to note that authorization ***does not*** refer to the ***access*** that the entities actually have to the TOE or its data. For example, suppose the owner of the system determines that no one other than employees were authorized to certain data on a TOE, yet they connect the TOE to the Internet. There are millions of entities that are not ***authorized*** to the data (because they are not employees), but they actually have connectivity to the TOE through the Internet and thus can attempt to access the TOE and its associated resources.

Entities are characterized according to the value of resources to which they are authorized; the extent of their authorization is implicitly a measure of how trustworthy the entity is with respect to compromise of the data (that is, compromise of any of the applicable security policies; e.g., confidentiality, integrity, availability). In other words, in this model the greater the extent of an entity's authorization, the more trustworthy (with respect to applicable policies) that entity is.

3.3 Selection of Appropriate Robustness level

Robustness is a characteristic of a TOE defining how well it can protect itself and its resources; a more robust TOE is better able to protect itself. This section relates the defining factors of IT environments, authorization, and value of resources to the selection of appropriate robustness levels.

When assessing any environment with respect to Information Assurance (IA) the critical point to consider is the likelihood of an attempted security policy compromise, which was characterized in the previous section in terms of entity authorization and resource value. As previously mentioned, robustness is a characteristic of a TOE that reflects the extent to which a TOE can protect itself and its resources. It follows that as the likelihood of an attempted resource compromise increases, the robustness of an appropriate TOE should also increase.

It is critical to note that several combinations of the environmental factors will result in environments in which the likelihood of an attempted security policy compromise is similar. Consider the following two cases:

- The first case is a TOE that processes only low-value data. Although the organization has stated that only its employees are authorized to log on to the system and access the data, the system is connected to the Internet to allow authorized employees to access the system from home. In this case, the least trusted entities would be unauthorized entities (e.g.,non-employees) exposed to the TOE because of the Internet connectivity. However, since only low-value data are being processed, the likelihood that unauthorized entities would find it worth their while to attempt to compromise the data on the system is low and selection of a basic robustness TOE would be appropriate.

- The second case is a TOE that processes high-value (e.g., classified) information. The organization requires that the TOE be stand-alone, and that every user with physical and logical access to the TOE undergo an investigation so that they are authorized to the highest value data on the TOE. Because of the extensive checks done during this investigation, the organization is assured that only highly trusted users are authorized to use the TOE. In this

case, even though high value information is being processed, it is unlikely that a compromise of that data will be attempted because of the authorization and trustworthiness of the users and once again selection of a basic robustness TOE would be appropriate.

The preceding examples demonstrate that it is possible for radically different combinations of entity authorization/resource values to result in a similar likelihood of an attempted compromise. As mentioned earlier, the robustness of a system is an indication of the protection being provided to counter compromise attempts. Therefore, a basic robustness system should be sufficient to counter compromise attempts where the likelihood of an attempted compromise is low. Figure 1 depicts the "universe" of environments characterized by the two factors discussed in the previous section: on one axis is the authorization defined for the least trustworthy entity, and on the other axis is the highest value of resources associated with the TOE.

As depicted in this figure, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflects the notion that different environments engender similar levels of "likelihood of attempted compromise," signified by a similar color. Further, the delineations between such environments are not stark, but rather are finely grained and gradual.

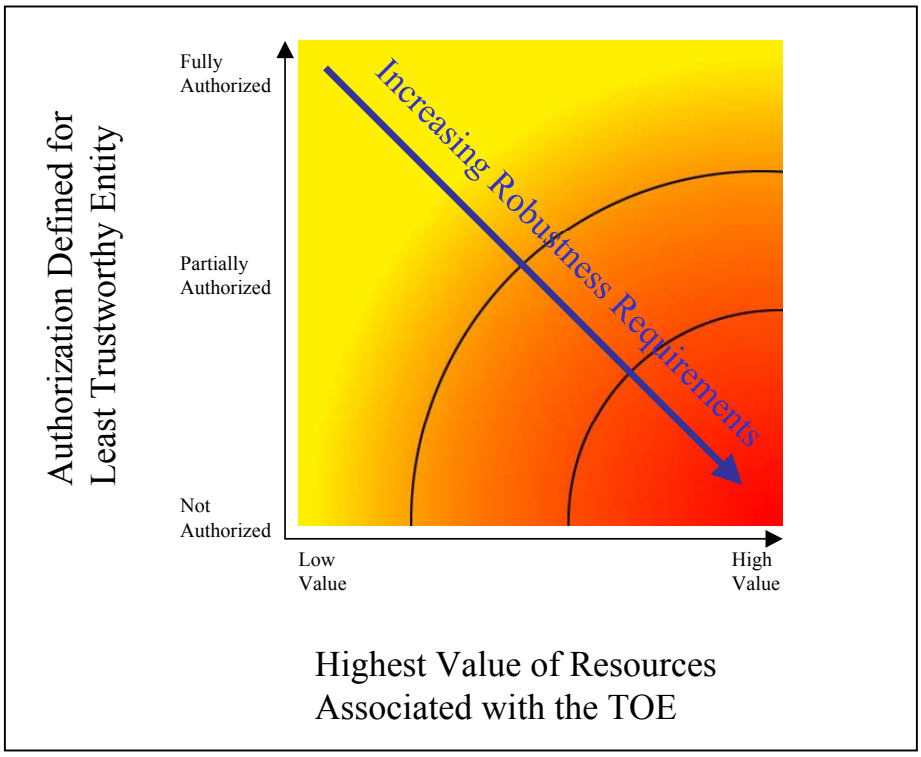

Figure 4 – Environmental Factors for Consideration

While it would be possible to create many different "levels of robustness" at small intervals along the “Increasing Robustness Requirements” line to counter the increasing likelihood of attempted compromise due to those attacks, it would not be practical nor particularly useful. Instead, in order to implement the robustness strategy where there are only three robustness levels: Basic, Medium, and High, the graph is divided into three sections, with each section corresponding to set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in Figure 4.

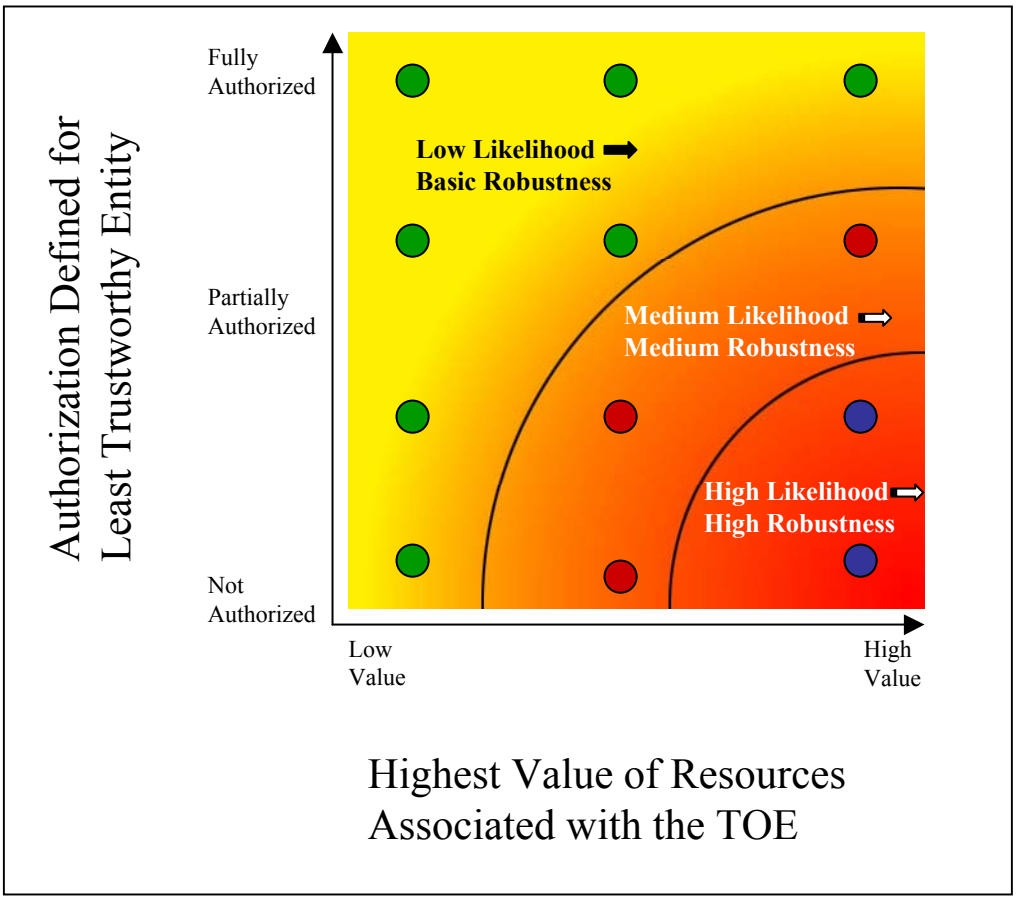

Figure 5 - Sectionalized Environments

In this second representation of environments and the robustness plane, the “dots” represent given instantiations of environments; like-colored dots define environments with a similar likelihood of attempted compromise. Correspondingly, a TOE with a given robustness should provide sufficient protection for environments characterized by like-colored dots. In choosing the appropriateness of a given robustness level TOE PP for an environment, then, the user must first consider the lowest authorization for an entity as well as the highest value of the resources in that environment. This should result in a “point” in the chart above, corresponding to the likelihood that that entity will attempt to compromise the most valuable resource in the environment. The appropriate robustness level for the specified TOE to counter this likelihood can then be chosen.

The difficult part of this activity is differentiating the authorization of various entities, as well as determining the relative values of resources; (e.g., what constitutes "low value" data vs. "medium value" data). Because every organization will be different, a rigorous definition is not possible. In Section 3.5 of this PP, the targeted threat level for a basic robustness Authorization Server TOE is characterized. This information is provided to help organizations insure that the functional requirements specified by this basic robustness PP are appropriate for their intended application of a compliant Authorization Server.

Basic robustness TOEs falls in the upper left area of the previously discussed robustness figures. A Basic Robustness TOE is considered sufficient for low threat environments or where compromise of protected information will not have a significant impact on mission objectives. This implies that the motivation of the threat agents will be low in environments that are suitable for TOEs of this robustness. In general, basic robustness results in "good commercial practices" that counter threats based in casual and accidental disclosure or compromise of data protected by the TOE.

Threat agent motivation can be considered in a variety of ways. One possibility is that the value of the data process or protected by the TOE will generally be seen as of little value to the adversary (i.e., compromise will have little or no impact on mission objectives). Another possibility, (where higher value data is processed or protected by the TOE) is that procuring organizations will provide other controls or safeguards (i.e., controls that the TOE itself does not enforce) in the fielded system in order to increase the threat agent motivation level for compromise beyond a level of what is considered reasonable or expected to be applied.

The remainder of this section addresses the following:

- Assumptions about the security aspects of a compliant TOE environment;
- Threats to TOE assets or to the TOE environment which must be countered; and
- Organizational security policies that compliant TOEs must enforce.

3.4 Authorization Server TOE Environment

The typical Authorization Server TOE deployment environment is unlike other information-technology-related TOEs in a number of respects:

Almost all "users," (i.e., people who receive access control support and indirectly interact with the TOE), are not operators of the system. Their only role is to present their identification (PKI certificate or user name / password) to a web server or application they desire access to. The web server agent or the application is the entity that actually interacts on behalf of the user with the Authorization Server components that contain high value information.

Authorization Server administrators are generally only a very small percentage of an organization population. These personnel are responsible for establishing the access control rules for high value data and have access to user sensitive personal attributes (i.e.: clearance information (in the case of Government operations) or payroll/personnel data (in the business

environment). Since the numbers are limited, only the most trustworthy personnel in a organization are generally assigned to administer the system. These administrator would normally consist of IT department administrators, web masters/data owners, and security professional who already have access to the information they are trying to protect. Therefore, even though high value information is being processed, it is unlikely that a compromise of that data will be attempted because of the authorization and trustworthiness of the actual administrative users.

Authorization Server vendors typically recommended a "Defense in Depth" strategy for the deployment of sensitive Authorization Server components. Regular users who desire to obtain access to an organization's protected web servers could be on the organization's internal operational "intranet" or on an external network that has limited access though the organizations gateway to its business partners. The protected web resources could be located on the organization's intranet, however, they are more commonly housed in a "data center" that again has further restricted access through products like packet filtering firewalls. It is in this more restrictive data center where the main Authorization Server Policy Engines are generally housed. For the most valuable data, and access to that data through the Authorization Servers Administrative Server, vendors recommend a private security enclave be deployed with another firewall to limit access to only authorized system administers and the other authorization server components. Figure 5 depicts the typical deployment environment for the deployment of Authorization Server components.

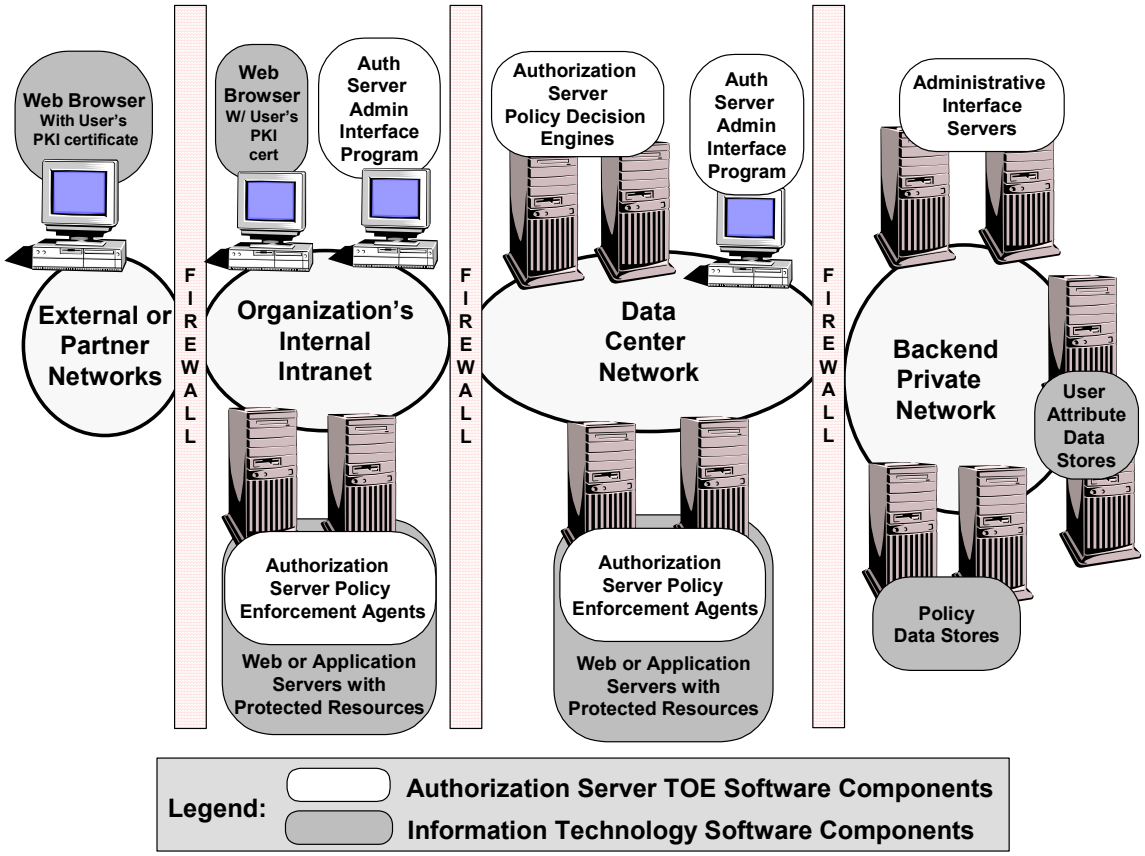

Figure 6 - Authorization Server Typical Deployment Environment

Figure 5 above also shows the Authororization Server administrative workstation which contain the Administrative Iterface Program ("client" or web browser"). These could be placed on either the data center or operational intranet depending on the organizations policy. The firewalls and the authoriztion server's own access control mechanism can restrict access to the Adinistrative Interface Server to only authorized personnel.

Figure 5 also depicts the need for multiple instances of both Authorization Server software components as well as some of the other IT software components. This will be a required to meet the high availability requirements of the PP. When a user attempts to gain access to a protected resource, the web server agent will attempt to contact the Authorization Server Policy Engine. If that component is not available, the Agent software must be able to contract another Policy Engine for service. The Policy Engine must also be able to contact the User Attribute and Policy Data Stores. However, since those software products are not part of the Authorization Server TOE, the IT Environment must ensure redundancy is provided to those components. This can be accomplished in a variety of ways with either high availability database software or with hardware load balancer and multiple directories or database. The implantation of that is beyond the scope of the Authorization Server TOE.

Since this is a "software only" Protection Profile, the Authorization Server environment will be expected to provide the hardware, operating system, and other security components (e.g.: time stamping, integrity checking) necessary to meet all the requirements to field a Basic Robustness system. Although the TOE will be responsible for generating audit data, the operating system or separate hardware/software components of the IT environment must provide the audit review, storage, and analysis required to meet the all the "basic robustness" requirements.

3.5 Assumptions

This section contains assumptions regarding the security environment and the intended usage of the TOE. The specific conditions identified in Table 1 are assumed to exist in a PP-compliant TOE environment.

Table 1 – TOE Assumptions

IDENTIFICATION	DESCRIPTION
A.CAPP_OS	The operating system the TOE operates on top of must be compliant with the Controlled Access Protection Profile.
A.COMMS	Adequate communications exist between the TOE components (internally) and between the TOE components and the IT components.
A.DYNMIC	The TOE will be managed in a manner that allows it to appropriately address changes in the IT System.
A.IT_ACCESS	The TOE has access to all the IT System data it needs to perform its functions.

IDENTIFICATION	DESCRIPTION
A.LOWEXP	The threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NOEVIL	Administrators are non-hostile, appropriately trained and follow all administrator guidance.
A.NO_GENERAL_ PURPOSE	There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the hardware platforms that the TOE administrative and authorization policy engine software are installed. This assumption does not apply to the Authorization Server "Agent" software.
A.NO_TOE_BYPASS	Users cannot gain access to recourses protected by the TOE without passing through the TOE access control mechanisms.
A.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.
A.SCALABLE	The TOE environment is appropriately scalable to provide support to the IT Systems in the organization it is deployed.
A.TOE_ENVIRONMENT_ ACCESS	The TOE environment will provide mechanisms that control a user's logical access to the TOE environmental components.

3.6 Threats

In addition to helping define the robustness appropriate for a given environment, the threat agent is a key component of the formal threat statements in the PP. Threat agents are typically characterized by a number of factors such as ***expertise***, ***available resources***, and ***motivation***. Because each robustness level is associated with a variety of environments, there are corresponding varieties of specific threat agents (that is, the threat agents will have different combinations of motivation, expertise, and available resources) that are valid for a given level of robustness. The following discussion explores the impact of each of the threat agent factors on the ability of the TOE to protect itself (that is, the robustness required of the TOE).

The ***motivation*** of the threat agent seems to be the primary factor of the three characteristics of threat agents outlined above. Given the same expertise and set of resources, an attacker with low motivation may not be as likely to attempt to compromise the TOE. For example, an entity with no authorization to low value data none-the-less has low motivation to compromise the data; thus a basic robustness TOE should offer sufficient protection. Likewise, the fully authorized user with access to highly valued data similarly has low motivation to attempt to compromise the data, thus again a basic robustness TOE should be sufficient.

Unlike the motivation factor, however, the same can't be said for ***expertise***. A threat agent with low motivation and low expertise is just as unlikely to attempt to compromise a TOE as an attacker with low motivation and high expertise; this is because the attacker with high expertise does not have the motivation to compromise the TOE even though they may have the expertise to do so. The same argument can be made for ***resources*** as well.

Therefore, when assessing the robustness needed for a TOE, the motivation of threat agents should be considered a "high water mark." ***That is, the robustness of the TOE should increase as the motivation of the threat agents increases.***

Having said that, the relationship between expertise and resources is somewhat more complicated. In general, if resources include factors other than just raw processing power (money, for example), then expertise should be considered to be at the same "level" (low, medium, high, for example) as the resources because money can be used to purchase expertise. Expertise in some ways is different, because expertise in and of itself does not automatically procure resources. However, it may be plausible that someone with high expertise can procure the requisite amount of resources by virtue of that expertise (for example, hacking into a bank to obtain money in order to obtain other resources).

It may not make sense to distinguish between these two factors; in general, it appears that the only effect these may have is to lower the robustness requirements. For instance, suppose an organization determines that, because of the value of the resources processed by the TOE and the trustworthiness of the entities that can access the TOE, the motivation of those entities would be "medium." This normally indicates that a medium robustness TOE would be required because the likelihood that those entities would attempt to compromise the TOE to get at those resources is in the "medium" range. However, now suppose the organization determines that the entities (threat agents) that are the least trustworthy have no resources and are unsophisticated. In this case, even though those threat agents have medium motivation, the likelihood that they would be able to mount a successful attack on the TOE would be low, and so a basic robustness TOE may be sufficient to counter that threat.

It should be clear from this discussion that there is no "cookbook" or mathematical answer to the question of how to specify exactly the level of motivation, the amount of resources, and the degree of expertise for a threat agent so that the robustness level of TOEs facing those threat agents can be rigorously determined. However, an organization can look at combinations of these factors and obtain a good understanding of the likelihood of a successful attack being attempted against the TOE. Each organization wishing to procure a TOE must look at the threat factors applicable to their environment; discuss the issues raised in the previous paragraph; consult with appropriate accreditation authorities for input; and document their decision regarding likely threat agents in their environment. The important general points we can make are:

- The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE**.**

- A threat agent's expertise and/or resources that are "lower" than the threat agent's motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however**).**

- The availability of attacks associated with high expertise and/or high availability of resources (for example, via the Internet or "hacker chat rooms") introduces a problem when trying to define the expertise of, or resources available to, a threat agent.

3.6.1 Threats Addressed by the TOE

Table 2 identifies the threats for the TOE and the IT Environment. The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed level of expertise of the attacker for all the threats is unsophisticated.

Table 2 – TOE Threats

IDENTIFICATION	DESCRIPTION
T. ACCIDENTAL_ ADMIN_ ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ACCIDENTAL_ CRYPTO_ COMPROMISE	A administrative user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.ACCIDENTAL_AUDIT_ COMPROMISE	An administrative user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

IDENTIFICATION	DESCRIPTION
	TOE resources from one user or process to another.
T.TSF_COMPROMISE	A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).
T.UNATTENDED_ SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ ACCESS	A user may gain access to user data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ ACTIONS	The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

3.7 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. This section identifies the organizational security policies applicable to the ASPP.

This PP was developed to meet the following policy guidance for Authorization Server systems that are deployed within the U.S Intelligence Community:

- Director, Central Intelligence Directive (DCID) 6/3
- Protection Level: 2
- Integrity Level of Concern: High
- Availability Level of Concern: High

The organizational security policy statements are derived from the following reference for Authorization Servers that are deployed within the DoD environment:

- DODD 5200.28, Security Requirements for Automated Information Systems

All PP-compliant TOEs must address the organizational security policies described in Table 3.

Table 3 – TOE Policies

IDENTIFICATION	DESCRIPTION
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.
P.ACCOUNTABILITY	The TOE shall log all actions by authorized users such that the authorized users can be held accountable for their actions within the TOE

IDENTIFICATION	DESCRIPTION
P.CRYPTOGRAPHY	Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e., encryption, decryption, signature, hashing, key exchange, and random number generation services).
P.HIGH_AVAILABILITY	The TOE shall meet the software related requirements of the DCID 6/3 for a "High" Level of Concern (LOC) in Availability. This includes providing resource allocations to support priority of service and fault tolerance.
P.RATINGS_MAINTENANCE	Procedures to maintain the TOE's rating must be in place, and these procedures must be implemented to maintain the TOE's rating once it is evaluated.

4 SECURITY OBJECTIVES

This chapter describes the security objectives for the TOE and the TOE's operating environment. The security objectives are divided between TOE Security Objectives (i.e., security objectives addressed directly by the TOE) and Security Objectives for the Operating Environment (i.e., security objectives addressed by the IT domain or by non-technical or procedural means).

The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.1 TOE Security Objectives

Table 4 defines the security objectives that are to be addressed by the TOE.

Table 4 – TOE Security Objectives

IDENTIFICATION	DESCRIPTION
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security-relevant events associated with users.
O.CONFIGURATION_ IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.
O.CORRECT_ TSF_OPERATION	The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
O.CRYPTOGRAPHY	The TOE shall use NIST FIPS 140-2 validated cryptographic services.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE to administrative users.
O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.
O.PARTIAL_FUNCTIONAL_ TESTING	The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.
O.PARTIAL_SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.

IDENTIFICATION	DESCRIPTION
O.RATINGS_ MAINTENANCE	Procedures to maintain the TOE's rating will be documented and followed.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.
O.FAULT_TOLERANCE	The TOE will provided limited capabilities to support degraded fault tolerance and fail over for some TOE components
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.
O.VULNERABILITY_ ANALYSIS	The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.

4.2 Security Objectives for the Operating Environment

Since this is an application "software only" PP, there are several objectives that must be met by the hardware components and the underlying operating systems to provide a secure TOE Environment. These include objectives that levy IT requirements on the hardware and operating system and those that can be satisfied by procedural or administrative measures.

Table 5 defines the security objectives that are to be addressed by the IT domain or by non-technical or procedural means. All of the assumptions stated in Section 3.5 are considered to be security objectives for the environment. There is an additional objective for the environment, OE.CRYPTANALYTIC. The mapping and rationale for the security objectives are described in Section 6.

Table 5 – TOE Operating Environment Security Objectives

IDENTIFICATION	DESCRIPTION
OE.CAPP_OS	Operating systems the TOE operates on top of will be compliant with the Controlled Access Protection Profile. The operating system will therefore provide all the capabilities outlined in the CAPP security function requirements and assurance requirements.
OE.COMMS	Sites deploying the TOE will provide adequate communications exist between the TOE components (internally) and between the TOE components and the IT components.
OE.CRYPTOGRAPHY	The TOE environmental components shall use NIST FIPS 140-2 validated cryptographic services.
OE.DISPLAY_BANNER	The underlying operating system of the TOE will display an advisory warning regarding use of the TOE to administrative users logging on the platform where the TOE software is installed.
OE.IT_ACCESS	Sites deploying the TOE will ensure the TOE has access to all the IT System data it needs to perform its functions.
OE.LOWEXP	Site deploying the TOE will establish a protective environment where the threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.
OE.MANAGE	The TOE environmental components will provide all the functions, facilities and competent individuals necessary to support the administrators in their management of the security of the environment, and restrict these functions and facilities from unauthorized use.
OE.NO_EVIL	Sites using the TOE shall ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.NO_GENERAL_ PURPOSE	There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the hardware platforms that the TOE administrative and authorization policy engine software are installed. This objective does not apply to the Authorization Server “Agent” software.
OE.NO_TOE_BYPASS	Users cannot gain access to recourses protected by the TOE without passing through the TOE access control mechanisms.
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.
OE.SCALABLE	Sites using the TOE will deploy the appropriate hardware and software environment to ensure the TOE system is scalable to provide support to the IT Systems in the organization it is deployed.
OE.TOE_ENVIRONMENT_ ACCESS	The TOE environment will provide mechanisms that control a user’s logical access to the environmental components.

5 IT SECURITY REQUIREMENTS

This section provides functional and assurance requirements that must be satisfied by a Protection Profile-compliant TOE. It also provides functional requirements the IT environment be must meet to deploy an Authorization Server in a secure manner, meeting the policy objectives. These requirements consist of functional components from Part 2 of the CC 3 and assurance components from Part 3 of the CC.

5.1 TOE Functional Security Requirements

This section provides functional and assurance requirements that must be satisfied by a PP-compliant TOE. These requirements consist of functional components from Part 2 with NIAP interpretations. Table 6 summarizes the TOE Functional Requirements to meet the stated objectives. Table 6 identifies the explicit requirements that were necessary to express the desired functionality or meet the NIAP Basic Robustness Consistency Guidance.

Table 6 – TOE Security Functional Requirements

FUNCTIONAL SECURITY CLASS	FUNCTIONAL SECURITY REQUIREMENT COMPONENTS
Audit Data Generation	FAU_GEN.1-NIAP-0410
User Identity Association	FAU_GEN.2-NIAP-0410
Cryptographic Key Generation	FCS_CKM.1
Cryptographic Key Distribution	FCS_CKM.2
Cryptographic Key Destruction	FCS_CKM.4
Cryptographic Operation (Encryption/Decryption AES)	FCS_COP.1
Access Control Policy	FDP_ACC.1
Access Control Functions	FDP_ACF.1
Full Residual Information Protection	FDP_RIP.2
Authentication Failure Handling	FIA_AFL.1-NIAP-0425
User Attribute Definition	FIA_ATD.1
Verification of Secrets	FIA_SOS.1
Timing of Authentication	FIA_UAU.1
Multiple Authentication Mechanisms	FIA_UAU.5

FUNCTIONAL SECURITY CLASS	FUNCTIONAL SECURITY REQUIREMENT COMPONENTS
Timing of Identification	FIA_UID.1
User-Subject Binding	FIA_USB.1.1-NIAP-0351
Management of Security Functions Behavior	FMT_MOF.1
Management of Security Attributes	FMT_MSA.1
Secure Security Attributes	FMT_MSA.2
Static Attribute Initialization	FMT_MSA.3-NIAP-409
Management of TSF Data	FMT_MTD.1
Management of Limits on TSF Data	FMT_MTD.2
Security Roles	FMT_SMR.1
Inter-TSF Confidentiality During Transmission	FPT_ITC.1
Internal TOE TSF Data Transfer	FPT_ITT.1
Non-bypassability of the TSP	FPT_RVM.1
TSF Domain Separation	FPT_SEP_EXP.1
TSF Domain Separation	FPT_SEP_EXP.2
TSF Testing	FPT_TST.EXP1.1
Degraded Fault Tolerance	FRU_FLT.1
Per User Attribute Limitation on Multiple Concurrent Sessions	FTA_MCS.2
Default TOE Access Banners	FTA_TAB.1

5.1.1 Class FAU: Security Audit

FAU_GEN.1-NIAP-0460 Audit Data Generation

The TSF shall be able to generate an audit record of the following auditable events:

Start-up and shutdown of the audit functions;

All auditable events for the *basic* level of audit **as identified in Table 7**;

[selection chose one of: [assignment: events at a basic level of audit introduced by the inclusion of additional SFRs determined by the ST author], [assignment: events commensurate with a basic level of audit introduced by the inclusion of explicit requirements determined by the ST author], "no additional events"].

Application Note: For the selection, the ST author should choose one of the assignments (as detailed in the following paragraphs), or select "no additional events".

For the first assignment, the ST author augments the table (or lists explicitly) the audit events associated with the basic level of audit for any SFRs that the ST author includes that are not included in this PP.

Likewise, for the second assignment the ST author includes audit events that may arise due to the inclusion of any explicit requirements not already in the PP. Because "basic" audit is not defined for such requirements, the ST author will need to determine a set of events that are commensurate with the type of information that is captured at the basic level for similar requirements.

If no additional (CC or explicit) SFRs are included, or if additional SFRs are included that do not have "basic" audit associated with them, then it is acceptable to assign "no additional events" in this item

FAU_GEN.1.2-NIAP-0460 - The TSF shall record within each audit record at least the following information:

a. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of below*].

Application Note: In column 3 of the Table 7, "if applicable" is used to designate data that should be included in the audit record if it "makes sense" in the context of the event that generates the record. If no other information is required (other than that listed in "a") for a particular audit event type, then an assignment of "none" is acceptable.

Dependency: FPT_STM.1 Reliable Time Stamps

Table 7 – Auditable Events

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL AUDIT RECORD CONTENTS
FCS_COP.1	Failure of cryptographic operation	Type of cryptographic operation Any applicable cryptographic mode(s) of operation, excluding any sensitive information

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL AUDIT RECORD CONTENTS
FDP_ACF.1	All requests to perform an operation on an object covered by the SFP.	The specific security attributes used in making an access check.
FIA_AFL.1.1-NIAP-0425	The reaching of the threshold for the unsuccessful authentication attempts.	The claimed identity of the user attempting to gain access
FIA_AFL.1.2-NIAP-0425	The actions (e.g.,disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g.,re-enabling of a terminal).	The claimed identity of the user attempting to gain access
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret	Identification of any changes to the defined quality metrics.
FIA_UAU.1	All use of the authentication mechanism;	All TSF mediated actions performed before authentication of the user.
FIA_UAU.5	The result of each activated mechanism together with the final decision.	Claimed identity of user being authenticated
FIA_UID.1	All use of the user identification mechanism, including the user identity provided.	Claimed identity of the user using the identification mechanism
FIA_USB.1.1-NIAP-0351	Success and failure of binding of user security attributes to a subject (e.g., success and failure to create a subject).	Identity of user whose attributes are attempting to be bound
FMT_MOF.1	All modifications in the behavior of the functions in the TSF.	Identity of administrator making the modifications
FMT_MSA.1	All modifications of the values of security attributes.	Identity of administrator making the modifications
FMT_MSA.2	All offered and rejected values a security attribute	All offered and accepted secure values for a security attribute.
FMT_MSA.3-NIAP-0409	All modifications of the values of static security attributes	Identity of administrator making the modifications

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL AUDIT RECORD CONTENTS
FMT_MTD.1	All modifications to the values of TSF data.	Identity of administrator making the modifications
FMT_MTD.2	All modifications of the limits All Modifications un the actions to be taken in case of violation of limits	Identity of administrator making the modifications
FPT_RCV.NIAP-0389-1	The fact that a failure or service discontinuity occurred; the resumption of the regular operation;	Type of failure or service discontinuity.
FPT_TST.EXP1.1	Execution of the TSF self tests and the results of the tests.	
FRU_FLT.1	Any failure detected by the TSF. Plus all TOE capabilities being discontinued due to a failure.	Identity of component that failed
FTA_MCS.2	Rejection of a new session based on the limitation of multiple concurrent sessions.	Capture of the number of currently concurrent user sessions and the user security attribute(s) (the identity of administrator)
FTA_TAB.1	None	

FAU_GEN.2-NIAP-0410 User Identity Association

FAU_GEN.2.1-NIAP-0410 – For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Application Note: For failed login attempts no user association is required because the user is not under TSF control until after a successful identification/authentication. User in this requirement is the userid for authorized users.

5.1.2 Cryptographic Support (FCS)

The cryptographic requirements are structured to support the implementation of FIPS 140-2 (Level 1) approved cryptographic algorithms and "best commercial practices" to protect communications between TOE components of distributed subsystems which are physically distributed on different hosts. Transport-Layer Security (TLS v1.0) is the most common means of securing connections between separated parts of the TOE in COTS software, and TLS can be configured to use solely FIPS 140-2 (Level 1) approved cryptographic algorithms. Additionally,

since the user logical interface to the TOE (for X.509 certificate based I&A) is considered to be the TLS connection between browser and web server, then the range of permitted cryptographic algorithms for that functions will include those FIPS-approved algorithms actually implemented in TLS by COTS browsers since that is out of the control of the TOE.

FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1 – At a minimum, the TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [Triple Data Encryption Standard (3DES) as specified in FIPS PUB 46-3 and implementing any mode of operation specified in FIPS PUB 46-3 with Keying Option 1 (K1, K2, K3 are independent keys)] and specified cryptographic key sizes [that are at least 192 binary digits in length] that meet the following: [FIPS PUB 46-3 with Keying Option 1 and FIPS PUB 140-2 (Level 1)].

Dependency: FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2.1 . The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [performed by commercially available Internet Key Exchange (IKE) implementations] that meets the following: [FIPS PUB 140-2 (Level 1) and ANSI X9-17].

Dependencies: FCS_CKM.1 – Cryptographic key generation
FCS_CKM.4 – Cryptographic key destruction
FMT_MSA.2 – Secure security attributes

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 - The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [which zeroizes all plaintext cryptographic keys and other unprotected security parameters within the device] that meets the following: [FIPS PUB 140-2, Security Level 1].

Dependencies: FCS_CKM.1 – Cryptographic key generation
FMT_MSA.2 – Secure security attributes

FCS_COP.1 Cryptographic operation

FCS_COP.1.1 - The TSF shall perform [encryption, decryption, and secure hash of network traffic as defined in the TOE security policy] in accordance with a specified cryptographic algorithm: [Triple Data Encryption Standard (3DES) as specified in RFC 2451 and implementing any mode of operation specified in FIPS PUB 46-3 with Keying Option 1 (K1, K2, K3 are independent keys)] and cryptographic key sizes [that are at

least 192 binary digits in length] that meet the following: [FIPS PUB 46-3 with Keying Option 1 and FIPS PUB 140-2 (Level 1) and HMAC-SHA-1-96 within ESP and AH (RFC 2404)].

Dependencies: FCS_CKM.1 – Cryptographic key generation
FCS_CKM.4 – Cryptographic key destruction
FMT_MSA.2 – Secure security attributes

Application Note: Triple DES encryption must protect all communications between the Authorized Administrator and the TOE and the communications between TOE components. The associated cryptographic module(s) must comply at a minimum with FIPS PUB 140-2 Level 1. A future migration to the Advanced Encryption Standard (AES) is anticipated when the national standards are established. The intent of this requirement is not for the evaluator to perform a FIPS PUB 140-2 evaluation; rather, the evaluator will check for a certificate, verifying that the module completed a FIPS PUB 140-2 evaluation.

5.1.3 User data protection (FDP)

FDP_ACC.1 Subset Access Control Policy

FDP_ACC.1.1 - The TSF shall enforce the [Authorization Server Access Control policy[on [assignment: list of subjects] acting on the behalf of users, [assignment: list of named objects], and [all the operations among subject and object covered by the Authorization Server Access Control policy.]

Application Note: There is not a single "standard" policy, like Discretionary Access Control (DAC). The Authorization Server Access Control Policy is based on the Security Administrator's defined rules or operations. The "subjects" are generally the Authorization Server "Agents." The "named objects" are the designated web based resources (web server, directories, files, or objects) that the Authorization Server is protecting.

Dependency: FDP_ACF.1 – Security attributed based access control

FDP_ACF.1-NIAP-0460 Security Attribute Based Access Control

FDP_ACF.1.1-NIAP-0460 –The TSF shall enforce the [Authorization Server Access Control policy] to objects based on the following:

a. [The user identity and group membership(s) associated with a subject; and

b. The time of day attribute associated with the operation; and

c. The following access control attributes associated with an object: List access control attributes. The attributes must provide permission attributes with:

- the ability to associate allowed or denied operations with one or more user identities;
- the ability to associate allowed or denied operations with one or more named group identities; and
- the ability to associate allowed or denied operations based on the time of day; and
- defaults for allowed or denied operations.]

Application Note: The Authorization Server software requires a mechanism that allows administrator to create rules or policies that mediate access control on the IT web resources it is protecting based on security attributes associated with subjects and objects. To clarify the intent of a requirement and identify implementation choices,the following type of permission attributes shouldbe supported:

Identity attribute may be associated with users, subjects, or objects,

Time attribute can be used to specify that access will be authorized during certain times of the day, or during certain days of the week,

Grouping attribute allows a single group of users to be associated with an operation for the purposes of access control. Specific "Roles" can also be expressed with a grouping attribute. ST developers can use the refinement operation to specify the maximum number of definable groups, the maximum membership of a group, and the maximum number of groups to which a user can concurrently be associated (if desired).

FDP_ACF.1.2-NIAP-0460 –The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

[a set of rules specifying the Authorization Server Access Control policy, where:

- The enforcement mechanisms (e.g., self/group/public controls, access control lists, communities of interest [COIs]) shall allow users to specify and control sharing of those objects by named individuals, or by defined groups of individuals, or by both, and shall provide controls to limit propagation of access rights.
- For each operation there shall be a rule, or rules, that use the permission attributes where the user identity of the subject matches a user identity specified in the access control attributes of the object, or
- For each operation there shall be a rule, or rules, that use the permission attributes where the group membership of the subject

matches a group identity specified in the access control attributes of the object, or

- For each operation there shall be a rule, or rules based on syntax with complex access equations be based on Boolean Logic (e.g.: is included, is not included, and, or, greater than, less than) that uses the permission attributes of individual or group membership of the subject and matches those to the appropriately specified access control attributes of the object, or

- For each operation there shall be a rule, or rules that use the permission attributes of individual or group membership of the subject, and matches those to the appropriately specified access control attributes of the object, where the access control attributes of the object are inherited hierarchically by default (resources (e.g., directories, files) deeper in a hierarchy structure inherit the access control attributes of the objects above them), or

- For each operation there shall be a rule, or rules that use the permission attributes of individual or group membership of the subject, and matches those to the appropriately specified access control attributes of the object, where the object's attributes can be specifically stated and are different from the permission attributes that the object inherits hierarchically by default, or

- For each operation there shall be a rule, or rules, that use the default permission attributes specified in the access control attributes of the object such that if all rules fail to provide a successful match the results is a denial of access.

FDP_ACF.1.3-NIAP-0460 –The TSF shall explicitly authorize access of subjects to objects based in the following additional rules: [access permission to an object by subjects not already possessing access permission shall only be assigned by authorized users.]

FDP_ACF.1.4-NIAP-0460 –The TSF shall explicitly deny access of subjects to objects based on the **[selection:[** assignment: rules, based on security attributes, that explicitly deny access of subjects to objects], "no additional explicit denial rules"].

Dependencies: FDP_ACC.1 – Subset access control
FMT_MSA.3 – Static attribute initialization

FDP_RIP.2 Full residual information protection

FDP_RIP.2.1 - The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all objects.

Application Note: The word "resource" is used numerous times in this PP to designate those elements outside the TOE that are protected by the TOE. However, following the required CC language, the word "resource" in the FDP-RIP.2.1 FSC refers to an internal TSF resource.

5.1.4 Identification and authentication (FIA)

TOE security functions implemented by a probabilistic or permutational mechanism (e.g., password or hash function) are required (at EAL2 and higher) to include a strength of function claim. Strength of Function shall be demonstrated for the authentication mechanism used by the administrators to be SOF-basic, as defined in Part 1 of the CC. Specifically, the local authentication mechanism must demonstrate adequate protection against attackers possessing a low attack potential.

FIA_AFL.1-NIAP-0425 Authentication failure handling

FIA_AFL.1.1-NIAP-0425 **-** The TSF shall detect when [a Security Administrator-configurable integer] of unsuccessful authentication attempts occur related to [administrators attempting to authenticate remotely to the TOE's administrative server, and authorized user entities requesting access to protected IT resources].

Application Note: This requirement also does not apply to the local administrators access to user accounts on the operating system that they use to gain local access to the TOE's administrative and configuration files. The IT environment should address access to an account on the operating system. Access to the TOE software via at least one local account should be allowed so as to avoid an administrative denial of service.

FIA_AFL.1.2-NIAP-0425 - When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [prevent the remote administrators, or

authorized IT entity from performing activities that require authentication until an action is taken by the Security Administrator].

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 – The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: list of security attributes as determined by the ST Author]

FIA_SOS.1 Specification and Verification of Secrets

FIA_SOS.1.1 **-** The TSF shall provide a mechanism to verify that secrets meet **[**the condition that passwords must contain a minimum of 8 alpha numeric characters with at least one numeric character**]**.

FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 - The TSF shall allow [verification of number of concurrent sessions established and auditing of attempted session establishment] on behalf of the **administrative** user to be performed before the **administrative** user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note: This requirement is needed due to the dependency of the multiple concurrent session control requirement (FTA_MCS.2). Since that requirement only applies to administrative users logging on to a session with the TOE, this requirement has been refined to reflect its applicability to administrative users only. For this requirement the "session" is defined the period of time the administrative user is logged onto a the TOE component. This does not preclude the use of HTPP/HTML interfaces which do not have a formal notation of a "session".

FIA_UAU_5 Multiple authentication mechanisms

FIA_UAU_5.1 **-** The TSF shall provide **support for the following authentication mechanisms [**user ID and password, X.509V3 identity certificates in software, and X.509V3 identity certificates from hardware tokens, [selection *other authentication mechanism(s) determined by the ST Author, or no additional authentication mechanism)***]** to support user authentication **before access to the TOE protected resources**.

Application Note: This to meet this requirement the ST must be capable of supporting authentication via user ID and password, X.509V3 identity certificate in software, and X.509V3 identity certificate from hardware tokens mechanism for user's requesting access to protected resources The ST author could chose to fill in the assignment with any additional authentication mechanism, such as a single-use authentication mechanism, if desired, or state no additional mechanisms

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to [the following rule: Group authenticators may only be used in conjunction with the use of an individual/unique authenticator, that is, individuals must be authenticated with an individual authenticator prior to use of a group authenticator].

FIA_UID.1 Timing of Identification

FIA_UID.1.1 Timing of Identification

The TSF shall allow **[**verification of number of concurrent sessions established and auditing of attempted session establishment**]** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note: All users, both those general users requesting access to a protected resource and administrative user, whether authenticated or not, will always be identified by providing a userid or digital certificate

FIA_USB.1 User-Subject Binding

FIA_USB.1.1-NIAP-0351 - The TSF shall associate all user security attributes with subjects acting on behalf of that user.

Application Note: User security attributes are defined in FIA_ATD.1.

Dependency: FIA_ATD.1 – User attribute definition

5.1.5 Security management (FMT)

FMT_MOF.1(1) - Management of security functions behavior (Account Management)

FMT_MOF.1.1(1) - The TSF shall restrict the ability to *enable, disable or modify the behavior of* the functions: [Account Management procedures that include identifying types of accounts (individual and group, conditions for group membership, associated privileges)] to [the Security Administrator].

Dependencies: FMT_SMR.1 Security Role

FMT_MOF.1(2) Management of security functions behavior (audit)

FMT_MOF.1.1(2) - The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions related to the security audit generation to the [Security Administrator].

Application Note: For the Audit function, enable and disable refer to the ability to enable or disable the audit mechanism as a whole. "Determine the behavior" means the ability to determine specifically what on the system is being audited, while "modify the behavior" means the ability to set or unset specific aspects of the audit mechanism, such as what user behavior is audited, etc.

Dependencies: FMT_SMR.1 Security Role

FMT_MSA.1.1 - The TSF shall enforce the [Authorization Server Access Control SPF] to restrict the ability to *query, modify or delete* the security attributes [associated with both users and protected resources which are used for access control permission rules] to [a designated Security Administrator].

Application Note: This requirement restricts management of the sensitive user attributes and the security attributes that make up a protected resources access control requirements (or rules) to a designated Security Administrator.

Dependency: FDP_ACC.1 Subset access control

FMT_MSA.2 Secure Security Attributes

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

Application Note: In addition to the functionality provided for cryptography to meet the dependency requirement foe FCS_CKM and FCS_COP, this requirement will also be used to prevent user authentication password reuse. A history of static authenticator changes should be maintained with assurance of non-replication of individual authenticators. When a user changing their password submits a previously used password, the system should consider that a "insecure" value for that security attribute and reject it.

Dependencies: ADV_SPM.1 Informal TOE security policy model
FDP_ACC.1 Subset access control

FMT_MSA.3-NIAP-0460 Static attribute initialization

FMT_MSA.3.1-NIAP-0460 –The TSF shall enforce the [Authorization Server Access Control SFP] to provide restrictive default values for security attributes that are used to enforce the SFP.

Application Note: "restrictive" in this case means that by default access is not authorized to a protected resource unless an explicit rule in the ruleset allows the access. By default, access to protected data is not allowed.

FMT_MSA.3.2 - The TSF shall allow the **[**the Security Administrator**]** to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attribute
FMT_SMR.1 – Security roles

FMT_MTD.1(1) Management of TSF data

FMT_MTD.1.1(1) - The TSF shall restrict the ability to change default, query, modify, delete, clear, [all TSF data, including system configuration files and cryptographic security data], to [the Security Administrators role].

Application Note: The ST author should iterate this requirement as necessary to ensure that the TSF data are characterized in terms of the functionality provided by the TOE, and that the access is appropriately restricted to the appropriate administrators and authorized IT entities. This requirement also restricts the ability to configure the TOE's cryptographic policy to the Security Administrator. Configuring the cryptographic policy is related to things such as: setting modes of operation, key lifetimes, selecting a specific algorithm, and key length.

Dependencies: FMT_MSA.1 Management of security attribute
FMT_SMR.1 – Security roles

FMT_MTD.1 Management of TSF data (Access Control policy ruleset)

FMT_MTD.1.1(2) – The TSF shall restrict the ability to *query, modify, delete, create*, [*selection: [assignment: other operations as determined by the ST Author], none*] the [access control policy rules] to [the Security Administrator].

Application Note: This restricts the specification of the access control policy ruleset identified in the FDP_ACC requirements to the Security Administrator. This specification is done using the attributes defined for those policies.

The ST author should fill in any TOE-specific operations that an administrator can perform on the ruleset in the assignment.

Dependencies: FMT_MSA.1 Management of security attribute
FMT_SMR.1 – Security roles

FMT_SMR.1 Restrictions on security roles

FMT_SMR.1.1 - The TSF shall maintain the roles: [Security Administrator; Audit Administrator; [selection: [assignment: and any other roles], none]

FMT_SMR.1.2 - The TSF shall be able to associate users with roles.

Dependency: FIA_UID.1 Timing of authentication

5.1.6 Protection of the TOE Security Functions (FPT)

FPT_ITC.1 Inter-TSF Confidentiality During Transmission

FPT_ITC.1.1 The TSF shall protect all TSF data transmitted from the TSF to a remote trusted IT product from unauthorized disclosure during transmission.

Application Note: This requirement ensures data exchanges are protected between the Authorization Server components and other trusted IT components, like the data stores (Directories or databases). Even though user attribute and policy (ruleset) data is stored in components outside the TOE boundary, it is considered "TSF data" since it is used by the TOE to make access control decisions. Data transmission should implement at least one of the following:

(1) Information distributed only within an area approved for open storage of the information.

(2) Information distributed via a Protected Distribution System (PDS). A PDS provides physical protection or intrusion detection for communications lines. A PDS can also provide need-to-know isolation for communications lines.

(3) Information distributed using NSA-approved encryption mechanisms appropriate for the classification of the information.

(4) Information distributed using a trusted courier.

FPT_ITT.1 Internal TOE TSF Data Transfer

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure and modification when it is transmitted between separate parts of the TOE**.**

Application Note: This requirement ensures data exchanges are protected between the

Authorization Server components.

FPT_RVM.1 Non-bypassability of the TSP

FPT_RVM.1.1 - The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_SEP.1 SFP domain separation

FPT_SEP_EXP.1 The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

FPT_SEP_EXP.2 The TSF shall enforce separation between the security domains of subjects in the TOE Scope of Control.

FPT_TST_EXP1.1 TSF testing

FPT_TST_EXP1.1.1 - The TSF shall provide administrator with the capability to verify the integrity of the following TSF data: [TOE system configuration files including cryptographically-related TSF data].

FPT_TST_EXP1.1.2 - The TSF shall provide administrator with the capability to verify the integrity of stored TSF executable code.

5.1.7 Resource Allocation

FRU_FLT.1 Degraded Fault Tolerance

FRU_FLT.1.1 - The TSF shall ensure the operation of **[authorization decisions** when the following failures occur: **[**assignment: single instance of authorization decision software fails**]**.

Application note: In the event of software failure of an authorization server policy engine, the web agents should have an automated failover capability allow access to an alternate authorization server policy engine, thereby continuing service.

Dependency: FPT_FLS.1 – Failure with preservation of secure state

5.1.8 TOE Access (FTA)

FTA_MCS.2 Per User Attribute Limitation on Multiple Concurrent Sessions

FTA_MCS.2.1 - The TSF shall restrict the maximum number of concurrent sessions that belong to the same **Administrative** user according to the rules [Security Administrator will determine maximum number of concurrent sessions].

FTA_MCS.2.2 - The TSF shall enforce, by default, a limit of **[**default number determined by Security Administrator**]** sessions per **Administrative** user.

Application note: This requirement is restricted to only administrative users since general users do not log on to the TOE with "sessions." General users can log onto multiple different web resources or applications which the TOE is providing authorization / access control support for, however, that is not considered a "session" with the TOE. All concurrent administrative sessions must be audited. For a distributed deployment scenario, with multiple Authorization Server Administration Interface Program concurrently active, the ST author can show compliance with FTA_MCS.2 requiring each instance of the Administration Interface to enforce this limitation independently, or by imposing a restriction across all concurrently active Administration Interfaces within a single installation.

Dependency: FIA_UID.1 – Timing of identification

FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1 -. Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorized use of the TOE.

Application Note: The access banner should appear before or in conjunction with the administrative users of the TOE being prompted for their user identification and authentication. General user's requesting access to protected web resources are not provided the banner. It would be the responsibility of the web server to provide the banner as part of the initial access to the web page. The intent of this requirement is to advise administrative users of warnings regarding the unauthorized use of the TOE and to provide the Security Administrator with control over what is displayed (e.g., if the Security Administrator chooses, they can remove banner information that informs the user of the product and version number).

5.2 Security Requirements for the IT Environment

This Protection Profile provides functional requirements for the IT Environment. Since this is a application "software only" PP, to deploy this software in a secure manner a significant amount of requirements must be met by the IT Environment. First, the software must be installed on a securely configured operating system that is complaint with the Control Access PP (CAPP) The CAPP OS will provide a security functionality to meet a wide range of IT objectives including Discretionary Access Control (DAC), audit services (including generation, protection, review and analysis), user Identification and Authentication, Self-Protection, and a reliable time stamping. A complete list of the CAPP OS security requirements is provided in the CAPP PP. The IT environment also includes authorized IT entities (e.g., the data stores (directory servers or

relational databases), the web servers (with files to be protected), a certificate authority server, NTP server) and any IT entities that are used by administrators to remotely administer the TOE (e.g., a workstation with a browser).

Table 8 summarizes the IT Environment Functional Requirements that are levied on IT Environment in addition to the CAPP compliant operating system requirements. These additional requirements are necessary to meet the stated objectives. Table 9 identifies the explicit requirements that were necessary to express the desired functionality or meet the NIAP Basic Robustness Consistency Guidance. The detailed explanation of these requirements is also provided below.

Table 8 - IT Environment Security Functional Requirements

IT Environment Functional Components (from CC Part 2 and NIAP Interpretations)	
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic key distribution
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1	Cryptographic operation
FDP_SDI.1	Stored Data Integrity Monitoring
FDP_UIT.1	Data exchange integrity
FPT_FLS.1	Failure with preservation of secure state
FPT_ITC.1	Inter-TSF confidentiality during transmission
FPT_RCV.1	Recovery to a Known State
FPT_RVM.1	Non-bypassability of the TSP
FRU_FLT.1	Degraded Fault Tolerance
FRU_PRS.2	Full Priority of Service
FRU_RSA.1(1)	Maximum quotas (transport-layer quotas)
FRU_RSA.1(2)	Maximum quotas (controlled connection-oriented quotas)
FTA_MCS.2	Per User Attribute Limitation on Multiple Concurrent Sessions
FTA_SSL.1	TSF-initiated Session Locking

IT Environment Functional Components (from CC Part 2 and NIAP Interpretations)	
FTA_SSL.2	User-initiated Locking
FTA_SSL.3	TSF-initiated Termination
FTA_TAB.1	Default TOE Access Banners

Table 9 – IT Environment Explicit Security Functional Requirements

IT Environment Explicit Functional Components	
FPT_SEP_ENV_EXP.1	TSF domain separation
FPT_TST.EXP1.1.1	TSF Testing
FPT_TST.EXP1.1.2	TSF Testing

5.2.1 Cryptographic Support (FCS)

The cryptographic requirements are structured to support the implementation of FIPS 140-2 (Level 1) approved cryptographic algorithms and "best commercial practices" to protect communications between TOE components and the IT environments which are under the direct control of the TOE installation (e.g.: a directory server acting as the UADS). Transport-Layer Security (TLS v1.0) is the most common means of securing connections between separated parts of the TOE in COTS software, and TLS can be configured to use solely FIPS 140-2 (Level 1) approved cryptographic algorithms. Additionally, since the user logical interface to the TOE (for X.509 certificate based I&A) is considered to be the TLS connection between browser and web server, then the range of permitted cryptographic algorithms for that functions will include those FIPS-approved algorithms actually implemented in TLS by COTS browsers since that is out of the control of the TOE.

FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1 – At a minimum, the IT environment shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [Triple Data Encryption Standard (3DES) as specified in FIPS PUB 46-3 and implementing any mode of operation specified in FIPS PUB 46-3 with Keying Option 1 (K1, K2, K3 are independent keys)] and specified cryptographic key sizes [that are at least 192 binary digits in length] that meet the following: [FIPS PUB 46-3 with Keying Option 1 and FIPS PUB 140-2 (Level 1)].

Dependency: FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2.1 . The IT environment shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [performed by commercially available Internet Key Exchange (IKE) implementations] that meets the following: [FIPS PUB 140-2 (Level 1) and ANSI X9-17].

Dependencies: FCS_CKM.1 – Cryptographic key generation
FCS_CKM.4 – Cryptographic key destruction
FMT_MSA.2 – Secure security attributes

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 - The IT environment shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [which zeroizes all plaintext cryptographic keys and other unprotected security parameters within the device] that meets the following: [FIPS PUB 140-2, Security Level 1].

Dependencies: FCS_CKM.1 – Cryptographic key generation
FMT_MSA.2 – Secure security attributes

FCS_COP.1 Cryptographic operation

FCS_COP.1.1 - The IT environment shall perform [encryption, decryption, and secure hash of network traffic as defined in the TOE security policy] in accordance with a specified cryptographic algorithm: [Triple Data Encryption Standard (3DES) as specified in RFC 2451 and implementing any mode of operation specified in FIPS PUB 46-3 with Keying Option 1 (K1, K2, K3 are independent keys)] and cryptographic key sizes [that are at least 192 binary digits in length] that meet the following: [FIPS PUB 46-3 with Keying Option 1 and FIPS PUB 140-2 (Level 1) and HMAC-SHA-1-96 within ESP and AH (RFC 2404)].

Dependencies: FCS_CKM.1 – Cryptographic key generation
FCS_CKM.4 – Cryptographic key destruction
FMT_MSA.2 – Secure security attributes

Application Note: Triple DES encryption must protect all communications between the authorized administrator and the TOE and the communications between TOE components. If the authorized administrator is using a COTS browser as part of their IT environment, then that browser should meet these security requirements. Additionally, if communications between TOE components relys on the underlying IT environment (e.g. a directory server) then at IT component must meet these cryptographic requirements. The associated cryptographic module(s) must comply at a minimum with FIPS PUB 140-2 Level 1. A future migration to the Advanced Encryption Standard (AES) is anticipated when the national standards are established. The intent of this requirement is not for the evaluator to perform a FIPS PUB 140-2 evaluation; rather, the evaluator will check for a certificate, verifying that the module completed a FIPS PUB 140-2 evaluation.

5.2.2 User data protection (FDP)

FDP_SDI.1 Stored Data Integrity Monitoring

FDP_SDI.1.1 - The IT environment shall monitor user data stored within the TSC for [integrity errors] on all objects, based on the following attributes: [software source / executable code storage and other user data attributes designated by the Security Administrator].

5.2.3 Protection of the TOE Security Functions (FPT)

FPT_FLS.1 Failure with Preservation of Secure State

FPT_FLS.1.1 - The IT environment shall preserve a secure state when the following types of failures occur: **[**assignment:

- Operating System software failure,
- Software failures to data stores components,
- Other IT Environment software components failures, and
- Software failures on interfaces between IT components and the TOE**]**.

FPT_ITC.1 Inter-TSF Confidentiality During Transmission

FPT_ITC.1.1 The IT environment shall protect all TSF data transmitted from the TSF to a remote trusted IT products from unauthorized disclosure during transmission.

Application Note: This requirement ensure data exchanges are protected between the Authorization Server components and other trusted IT components, like the data stores (directories or databases). Even though user attribute and policy (ruleset) data is stored in components outside the TOE boundary, it is considered "TSF data" since it is used by the TOE to make access control decisions.

FPT_RCV.NIAP-0389-1 Recovery to Known State

FPT_RCV.NIAP-0389-1.1 For [

- Operating System software failure,
- Software failures to data stores components,
- Other IT Environment software components failures, and
- Software failures on interfaces between IT components and the TOE**]**.

the IT environment shall ensure the return of the IT software components to a previously known state using automated procedures.

FPT_RCV.NIAP-0389-1.2 When automated recovery from a failure or service discontinuity is not possible, the IT environment shall enter a maintenance mode where the ability to return the IT software components to a previously known state is provided.

FPT_RVM.1 Non-bypassability of the TSP

FPT_RVM.1.1 - The IT environment shall ensure that TSP enforcement functions are invoked and succeed before each function within the IT environment's scope of control is allowed to proceed.

Application note: The IT Environment of the protected web resources should be locked down such that access to those resources are not permitted via bypassing the web server interface and the TOE web agent and going directly to the operating system file structure to obtain access to the resource.

FPT_SEP.ENV_EXP SFP domain separation

FPT_SEP_ENV_EXP.1 The IT environment shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

FPT_SEP_ENV_EXP.2 The TSF shall enforce separation between the security domains of subjects in the IT environment's Scope of Control.

FPT_TST_EXP2.1 IT environment testing

FPT_TST_EXP2.1.1 – The IT environment shall run a suite of self-tests *during initial start-up, periodically during normal operation as specified by the administrator*, and at the [request of an administrator] to demonstrate the correct operation of the IT component security function.

FPT_TST_EXP2.1.2 - The IT environment shall provide administrator with the capability to verify the integrity of the following IT component security function data: [assignment: system configuration files including cryptographically-related data for which integrity validation is required].

FPT_TST_EXP2.1.3 - The IT environment shall provide administrator with the capability to verify the integrity of stored TSF executable code.

5.2.4 Resource Allocation

FRU_FLT.1 Degraded Fault Tolerance

FRU_FLT.1.1 - The IT environment shall ensure the operation of **[**assignment:

- Hardware platforms for TOE components (except web agents)
- Operating Systems for TOE components (expect web agents)
- Data Stores (directories or databases) which the authorization server policy engine software request user and policy data from**]**

when the following failures occur: **[**:

- Loss of primary power to TOE and IT environment components
- Loss of access to single instance of the policy and user data stores due to hardware, software, or primary communications failure**]**.

Application note: To prevent a failure of the Authorization Server system in the event of a primary electrical power failure to the IT hardware that host the TOE and the related IT

environment components (data stores) should be on Uninterrupted Power Supplies (UPS) and connected to a secondary back up power source.

In the event of hardware or software failure of a single instance user or policy data store, the IT environment must provide a mechanism to route the authorization server policy engine's request for user and policy data to an alternative data store. This should be an automated fail over capability allow access to an backup data store, thereby continuing service. This can be accomplished via high availability software for databases and directories servers, or could be accomplished via hardware load balancing devices. The IT environments fail over solution should also be able to alternately route the request from the authorization server policy engine to an available data stores in the event of a primary communications failure preventing connection to the primary data store.

FRU_PRS.2 Full Priority of Service

FRU_PRS.2.1 The IT environment shall assign a priority to each subject in the IT environment security function.

FRU_PRS.2.2 The IT environment shall ensure that each access to all shareable resources shall be mediated on the basis of the subjects assigned priority.

FRU_RSA.1- Maximum quotas (transport-layer quotas)

FRU_RSA.1.1(1) - The IT environment shall enforce maximum quotas of the following resources: [transport-layer representation] that *users or other subjects* can use *over a specified period of time*.

Application Note: "transport-layer representation" refers specifically to the TCP SYN attack, where half-open connections are established thus exhausting the connection table resource. If the IT componnet does not implement the TCP/IP protocol, this requirement would apply to a similar type of transport-layer entity for that IT environment protocol stack.

5.2.5 TOE Access (FTA)

FTA_MCS.2 Per User Attribute Limitation on Multiple Concurrent Sessions

FTA_MCS.2.1 - The IT environment shall restrict the maximum number of concurrent sessions that belong to the same **Administrative** user according to the rules [rules for the number of maximum concurrent sessions are determined by Security Administrator].

FTA_MCS.2.2 - The IT environment shall enforce, by default, a limit of **[**assignment: default number determined by Security Administrator**]** sessions per user.

Application note: This requirement is restricted to only administrative users since general users do not log on to the IT environment components with "sessions."

Dependency: FIA_UID.1 – Timing of identification

FTA_SSL.1 TSF-initiated session locking

FTA_SSL.1.1 - The IT environment shall lock a local interactive session after [a Security Administrator-specified time period of inactivity] by:

- clearing or overwriting display devices, making the current contents unreadable.
- disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 - The IT environment shall require the user to re-authenticate prior to unlocking the session.

Dependency: FIA_UAU.1 – Timing of authentication

FTA_SSL.2 User-initiated locking

FTA_SSL.2.1 - The IT environment shall allow user-initiated locking of the user's own local interactive session by:

- clearing or overwriting display devices, making the current contents unreadable.
- disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 - The IT environment shall require the user to re-authenticate prior to unlocking the session.

Dependency: FIA_UAU.1 – Timing of authentication

FTA_SSL.3 IT environment-initiated termination

FTA_SSL.3.1 -The IT environment shall terminate a interactive session after a [Security Administrator-configurable time interval of session inactivity].

Application Note: The interactive sessions in FTA_SSL.1, FTA_SSL.2 and FTA_SSL.3 are those of the administrative users. Non-administrators do not have any interactive sessions with the IT environment components.

FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1 -. Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorized us of the TOE

Application Note: The access banner should appear before or in conjunction with the administrative users of the IT environment component being prompted for their user identification and authentication. The intent of this requirement is to advise administrative users of warnings regarding the unauthorized use of the IT environment component and to provide the Security Administrator with control over what is displayed (e.g., if the Security Administrator chooses, they can remove banner information that informs the user of the product and version number).

5.3 TOE Security Assurance Requirements

The TOE assurance requirements for this PP are EAL2 augmented. All assurance requirements are summarized in the table below. The augmented requirements are in bold print.

Table 10 – Assurance Requirements: EAL2 Augmented

Assurance Class	Assurance Components	
Configuration management	ACM_CAP.2	Configuration items
Delivery and operation	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation, and start-up procedures
Development	ADV_FSP.1	Informal functional specification

Assurance Class	Assurance Components	
	ADV_HLD.1	Descriptive high-level design
	ADV_RCR.1	Informal correspondence demonstration
	ADV_SPM.1	Informal TOE security policy model
Guidance documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life cycle support	**ALC_FLR.2**	**Flaw Reporting Procedures**
Tests	ATE_COV.1	Evidence of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability assessment	**AVA_MSU.1**	**Examination of Guidance**
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis

ACM_CAP.2 Configuration items

Developer action elements:

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2C The TOE shall be labeled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.

ACM_CAP.2.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.6C-NIAP-0412 - The CM system shall uniquely identify all configuration items.

Evaluator action elements:

ACM_CAP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_DEL.1 Delivery procedures

Developer action elements:

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

Evaluator action elements:

ADO_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1 Installation, generation, and start-up procedures

Developer action elements:

ADO_IGS.1.1D - The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

ADO_IGS.1.1C - The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

Evaluator action elements:

ADO_IGS.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E - The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

ADV_FSP.1 Informal functional specification

Developer action elements:

ADV_FSP.1.1D The developer shall provide a functional specification.

Content and presentation of evidence elements:

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

Evaluator action elements:

ADV_FSP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.1 Descriptive high-level design

Developer action elements:

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

Evaluator action elements:

ADV_HLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_RCR.1.1D - The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C - For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

ADV_RCR.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Application Note: The intent of this requirement is for the vendor to provide, and the evaluator to confirm, that there exists accurate, consistent, and clear mappings between each level of design decomposition. Thus there can be no TOE security functions defined at a lower layer of abstraction absent from a higher level of abstraction and vice versa.

ADV_SPM.1 Informal TOE security policy model

Developer action elements:

ADV_SPM.1.1D - The developer shall provide a TSP model.

ADV_SPM.1.2D - The developer shall demonstrate correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements:

ADV_SPM.1.1C - The TSP model shall be informal.

ADV_SPM.1.2C - The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3C - The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C - The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

Application Note: As part of the secure state, the cryptographic module is in a known state such that all critical areas are empty of plaintext/red/secret data and inaccessible to processes, and all security policies are enforced.

Evaluator action elements:

ADV_SPM.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_ADM.1 Administrator guidance

Developer action elements:

AGD_ADM.1.1D - The developer shall provide administrator guidance addressed to system administrative personnel.

Content and presentation of evidence elements:

AGD_ADM.1.1C - The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C - The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C - The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C - The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C - The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C - The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C - The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C - The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD_ADM.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_USR.1 User guidance

Developer action elements:

AGD_USR.1.1D - The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C - The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C - The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C - The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C - The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C - The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C - The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD_USR.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.2 Flaw Reporting Procedures

ALC_FLR.2.1D - The developer shall document the flaw remediation procedures.

ALC_FLR.2.2D - The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.1C - The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C - The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C - The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C - The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5C - The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.6C - The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_COV.1 Evidence of coverage

Developer action elements:

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

Content and presentation of evidence elements:

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

Evaluator action elements:

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

Developer action elements:

ATE_FUN.1.1D - The developer shall test the TSF and document the results.

ATE_FUN.1.2D - The developer shall provide test documentation.

Content and presentation of evidence elements:

ATE_FUN.1.1C - The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C - The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C - The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C - The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C - The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

ATE_FUN.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent testing - sample

Developer action elements:

ATE_IND.2.1D - The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

ATE_IND.2.1C - The TOE shall be suitable for testing.

ATE_IND.2.2C - The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E - The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E - The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

AVA_MSU.1 Examination of guidance

Developer action elements:

AVA_MSU.1.1D The developer shall provide guidance documentation.

Content and presentation of evidence elements:

AVA_MSU.1.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.1.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.1.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.1.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

Evaluator action elements:

AVA_MSU.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.1.2E The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.1.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_SOF.1 Strength of TOE security function evaluation

Developer action elements:

AVA_SOF.1.1D - The developer shall perform a strength of TOE security function analysis for each mechanism identified in the Security Target as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C - For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C - For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

AVA_SOF.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E - The evaluator shall confirm that the strength claims are correct.

AVA_VLA.1 Developer vulnerability analysis

Developer action elements:

AVA_VLA.1.1D The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2D The developer shall document the disposition of obvious vulnerabilities. Content and presentation of evidence elements:

AVA_VLA.1.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

Evaluator action elements:

AVA_VLA.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6 RATIONALE

This section describes the rationale for the Security Objectives and Security Functional Requirements as defined in Section 4 and Section 5, respectively. Additionally, this section describes the rationale for the strength of function (SOF) claim. Table 12 illustrates the mapping from Security Objectives to Threats and Policies.

6.1 Rationale for TOE Security Objective

Table 11 - Security Objectives to Threats and Policies Mappings

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.ACCIDENTAL_ADMIN_ ERROR: An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	**O.ADMIN_GUIDANCE**: The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE (ADO_DEL.1,ADO_IGS.1, AGD_ADM.1, AGD_USR.1, AVA_MSU.1) help to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is unsecure.
T.ACCIDENTAL_AUDIT_ COMPROMISE: A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	**OE.CAPP_OS**: The CAPP compliant OS will provide the capability to protect audit information. **OE.PHYSICAL** The environment must address the possible compromise of audit data due to physical means **O.RESIDUAL_ INFORMATION**: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **O.PARTIAL_SELF_PROTECTION**: The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. .	OE.CAPP_OS contributes to mitigating this threat by controlling access to the audit trail. No one is allowed to modify audit records, the Audit Administrator is the only one allowed to delete the audit trail. The operating system has the capability to prevent auditable actions from occurring if the audit trail is full. O.RESIDUAL_INFORMATION (FDP_RIP.2) prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a TOE resource (e.g., memory). By ensuring the TOE prevents residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data. O.PARTIAL_SELF_PROTECTION (FPT_SEP_EXP.1, FPT_SEP_EXP.2 FPT_RVM.1) contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain security domains of subjects in the TOE Scope of Control, it could not be trusted to control access to the resources under its control, which includes the audit trail.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.ACCIDENTAL_ CRYPTO_ COMPROMISE: A user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.	**O.RESIDUAL_ INFORMATION**: The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION (FCS_CKM.2, FCS_CKM.4) mitigates the possibility of malicious users or processes from gaining inappropriate access to cryptographic data, including keys. This objective ensures that the cryptographic data does not reside in a resource that has been used by the cryptographic module and then reallocated to another process.
T.MASQUERADE: A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	**O.TOE_ACCESS**: The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS (FIA_AFL.1-NIAP-0425, FIA_ATD.1, FIA_UID.1, FIA_UAU.1, FIA_UAU.2, AVA_SOF.1, FIA_SOS.1) mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.
T.POOR_DESIGN: Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.	**O.CONFIGURATION_IDENTIFICATION**: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly, **O.DOCUMENTED_DESIGN**: The design of the TOE is adequately and accurately documented. **O.VULNERABILITY_ANALYSIS**: The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.CONFIGURATION_IDENTIFICATION (ACM_CAP.2, ALC_FLR.2) contributes to this objective by requiring the developer have a configuration configuration item, a reference for each version of the TOE, and a Configuration Management (CM) system with CM documentation. The developer is also required to establish flaw remediation procedures for accepting and acting upon user reports of security flaws and ensuring that any reported flaws are corrected. O.DOCUMENTED_DESIGN (ADV_FSP.1, ADV_HLD.1, ADV_RCR.1) counters this threat, to a degree, by requiring that the TOE be developed using a documented design engineering approach. By providing at least a high level of informal documenting of the security mechanisms in the TOE, the design of the TOE can be understood, which increases the chances that design errors will be discovered. O.VULNERABILITY_ANALYSIS (AVA_VLA.1) ensures that the design of the TOE is analyzed by the developer for obvious design flaws. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.POOR_IMPLEMENTATION: Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.	**O.CONFIGURATION_IDENTIFICATION**: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.,	O.CONFIGURATION_IDENTIFICATION (ACM_CAP.2, ALC_FLR.2) contributes to this objective by requiring the developer have a configuration item, a reference for each version of the TOE, and a Configuration Management (CM) system with CM documentation. The developer is also required to establish flaw remediation procedures for accepting and acting upon user reports of security flaws and ensuring that any reported flaws are corrected. Following a good CM process during development will reduce the risk of a implementation errors.
	O.PARTIAL_FUNCTIONAL_TESTING: The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	O.PARTIAL_FUNCTIONAL_ TESTING (ATE_COV.1, ATE_FUN.1, ATE_IND.2) increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification and high level design) will be discovered through testing.
	O.VULNERABILITY_ANALYSIS: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS (AVA_VLA.1) ensures that the design of the TOE is analyzed for obvious design flaws buy the developer. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.POOR_TEST: Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.	**O.CORRECT_ TSF_OPERATION**: The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. **O.PARTIAL_FUNCTIONAL_TESTING**: The TOE will undergo some security functional testing that demonstrates the TSF satisfies the security functional requirements. **O.VULNERABILITY_ANALYSIS**: The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	Design analysis determines that TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded. The O.CORRECT_TSF_OPERATION (FPT_TST_EXP1.1.1, FPT_TST_EXP1.1.2) provides administrators with the capability to verify the integrity TSF data, including stored TSF executable code and configuration file. O.PARTIAL_FUNCTIONAL_TESTING (ATE_FUN.1, ATE_COV.1, ATE_IND.2) ensures that functional testing is performed to ensure the TSF satisfies the security functional requirements and demonstrates that the TOE's security mechanisms operate as documented. While functional testing serves an important purpose, it does not ensure the TSFI cannot be used in unintended ways to circumvent the TOE's security policies. O.VULNERABILITY_ANALYSIS (AVA_VLA.1) ensures that the design of the TOE is analyzed by the developer for obvious design flaws. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.
T.RESIDUAL_DATA: A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.	**O.RESIDUAL_INFORMATION**: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION (FDP_RIP.2) counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets will not have residual data from another packet due to the padding of a packet. This ensures successful access control decisions make for one user do not carry over to the next user.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.TSF_COMPROMISE: A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).	**O.RESIDUAL_INFORMATION**: The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated. **O.PARTIAL_SELF_PROTECTION**: The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. **O.MANAGE**: The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.RESIDUAL_INFORMATION (FDP_RIP.2) counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets will not have residual data from another packet due to the padding of a packet. This ensures successful access control decisions make for one user do not carry over to the next user. O.PARTIAL_SELF_PROTECTION (FPT_SEP_EXP.1, FPT_SEP_EXP.2 FPT_RVM.1, FPT_FLS.1, FPT_ITC.1, FPT_ITT.1, FPT_RCV.NIAP-0389-1, and ADC_SPM.1) contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain security domains of subjects in the TOE Scope of Control, it could not be trusted to control access to the resources under its control. It requires that the TSF be able to protect itself from tampering and that the security mechanisms in the TSF cannot be bypassed. It also requires software failures to a secure state and recovery to known state. This also provides protection for TSF data during transmission between TOE components and to remote trusted IT products. Without this objective, there could be no assurance that users could not view or modify TSF data or TSF executables. It provides protection by ensuring that the TOE does not continue to operate in an insecure state when a software failure occurs. O.MANAGE (FMT_MTD.1(1)-(2), FMT_MTD.2, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3.1-NIAP-0409, FMT_MOF.1 and FMT.SMR.1) is necessary in order to define an access control policy to control access to TSF data or the resources being protected by the TOE. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.
T.UNATTENDED_ SESSION: A user may gain unauthorized access to an unattended session.	**OE.TOE_ENVIRONMENT_ACCESS** The operating system will provide mechanisms that control a user's logical access to a TOE session.	OE.TOE_ENVIRONMENT_ACCESS (FTA_SSL.1. FTA_SSL.2, FTA_SSL.3, and AVA_SOF.1) helps to mitigate this threat by including mechanisms that place controls on user's sessions. Local administrator's sessions are locked and remote sessions are dropped after a Security Administrator defined time period of inactivity. Locking the local administrator's session reduces the opportunity of someone gaining unauthorized access the session when the console is unattended. Dropping the connection of a remote session (after the specified time period) reduces the risk of someone accessing the remote machine where the session was established, thus gaining unauthorized access to the session.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
T.UNAUTHORIZED_ ACCESS: A user may gain access to user data for which they are not authorized according to the TOE security policy.	**O.MEDIATE**: The TOE must protect user data in accordance with its security policy.	O.MEDIATE (FDP_ACC.1, FDP_ACF.1, and FDP_RIP.2) works to mitigate this threat by ensuring that all requests to access user data, or data being protected by the TOE, are subject to an Authorization Server access control policy. A TOE policy engine enforces rules to determine if an operation among controlled subjects and controlled objects is allowed based on the security attributes of the user and the object. The user attributes can be based on group membership (or roles), time of day, or other Boolean Logic expressions. The TOE requires successful authentication to the TOE prior to gaining access to administrative services on or mediated by the TOE to protected resources. Communications between the TOE components must be protected from unauthorized disclosure to ensure integrity and confidentiality of the user data. Lastly, the TSF must ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the Security Administrator. This feature ensures that no other user can modify the access control policy to bypass the intended TOE security policy.
T.UNIDENTIFIED_ACTIONS: The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.	**OE.CAPP_OS**: The operating system will provide the capability to selectively view audit information.	OE.CAPP_OS (which includes FAU_SAA.1-NIAP-0460, FAU_SAR.1, FAU_SAR.3) helps to mitigate this threat by providing the Security Administrator with be a set of rules for monitoring the audited events and based upon these rules can indicate a potential violation of the TSP. A required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events, when the Security or Audit Administrator reviews the audit records, they can determine the occurrences of these events (e.g. set number of authentication failures, etc.). A search and sort capability provides an efficient mechanism for the Audit Administrator to view pertinent audit information.
P.ACCESS_BANNER: The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.	**O.DISPLAY_BANNER**: The TOE will display an advisory warning regarding use of the TOE.	O.DISPLAY_BANNER (FTA_TAB.1) satisfies this policy by ensuring that the TOE displays a Security Administrator configurable banner that provides all users with a warning about the unauthorized use of the TOE.

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
P.ACCOUNTABILITY: The authorized users of the TOE shall be held accountable for their actions within the TOE.	**O.AUDIT_GENERATION**: The TOE will provide the capability to detect and create records of security-relevant events associated with users. **OE.CAPP_OS**: The IT Environment shall provide reliable time stamps and the capability for the administrator to set the time used for these time stamps. **O.TOE_ACCESS**: The TOE will provide mechanisms that control a user's logical access to the TOE.	O.AUDIT_GENERATION (FAU_GEN.1-NIAP-0460, FAU_GEN.2-NIAP-410, FIA_USB.1-NAIP-0351,) addresses this policy by providing the Security Administrator with the capability of configuring the audit mechanism to record the actions of a specific user.or OE.CAPP_OS (which includes FAU_SEL.1-NIAP-0407) supports the review the audit trail based on the identity of the user. The OE.CAPP_OS also includes FPT_STM.1 and FMT_MTD.1, plays a role in supporting this policy by requiring the IT environment to provide a reliable time stamp (configured locally by the Security Administrator or via an external NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred. O.TOE_ACCESS (FIA_UID.1 and FTA_MCS.2) supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or access to any TOE protected resource that the TOE is mediating access on behalf of the users.
P.CRYPTOGRAPHY: Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).	**O.CRYPTOGRAPHY**: The TOE shall use NIST FIPS 140-2 validated cryptographic services. **O.RESIDUAL_INFORMATION**: The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	O.CRYPTOGRAPHY (FCS_CKM.1FCS_CKM.2, FCS_CKM.4, FCS_COP.1) satisfies this policy by requiring the TOE to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to between software components of the TOE and for TSF data being transfer to/from trusted IT environment components. O.RESIDUAL_INFORMATION (FCS_CKM_EXP.2, FCS_CKM.4) satisfies this policy by ensuring that cryptographic data are cleared from resources that are shared between users. Keys must be zeroized according to FIPS 140-2 and the storage location for the keys must be overwritten three or more times upon the transfer of keys to another location.
P.HIGH_AVAILABILITY	**O.FAULT_TOLERANCE** The TOE will provide limited capabilities to support degraded fault tolerance and fail over for some TOE components.	O.FAULT_TOLERANCE (FRU_FLT.1) helps satisfy the policy by ensuring that when a single instance of authorization server policy engine fails, operations are continued by an alternate authorization server policy engine.

6.2 Rationale for the Security objectives and Security Functional Requirements for the Environment

Eleven of the security objectives for the environment are restatements of an assumption found in Section 3. Table 12 provides a mapping of the assumptions to the environmental security objectives.

Table 12 – Assumptions to Environment Security Objectives Mappings

Assumption	Environment Security Objective
A.CAPP_OS	OE.CAPP_OS
A.COMMS	OE.COMMS
A.IT_ACCESS	OE.IT_ACCESS
A.LOWEXP	OE.LOWEXP
A.MANAGE	OE.MANAGE
A.NOEVIL	OE.NO_EVIL
A.NO_GENERAL_ PURPOSE	OE.NO_GENERAL_ PURPOSE
A.NO_TOE_BYPASS	OE.NO_TOE_BYPASS
A.PHYSICAL	OE.PHYSICAL
A.SCALABLE	OE.SCALABLE
A.TOE_ENVIRONMENT_ ACCESS	OE.TOE_ENVIRONMENT_ ACCESS

There is one security objective for the environment, OE.DISPLAY_BANNER, that maps to the P.ACCESS_BANNER Policy. This objective is levied upon the operating system component.

There is one security objective for the environment, OE.RESOURCE_ALLOCATION that maps to the P.HIGH_AVAILABILITY Policy. This objective is levied upon the operating system components and hardware to provide resource allocations to support priority of service and fault tolerance.

The non-IT security objective OE.CRYPTANALYTIC is necessary to counter the threat T.CRYPTO_COMPROMISE and addresses the policy P.CRYPTOGRAPHY. This non-IT security objective ensures that the cryptographic methods used in the TOE have been evaluated and verified to be FIPS 140-2 compliant. This non-IT security objective maps to the environmental requirement FPT_ITC.1 ensuring that encryption is used on the communication channel between authorized IT entities and the TOE.

6.3 Rationale for TOE Security Requirements

Table 13 - Rationale for TOE Security Requirements

Objective	Requirements Addressing the Objective	Rationale
O. ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	ADO_DEL.1 ADO_IGS.1 AGD_ADM.1 AGD_USR.1 AVA_MSU.1	ADO_DEL.1 ensures that the administrator is provided documentation that describe all procedures that are necessary to maintain security when receiving and distributing versions of TOE software at a user's site. The ADO_IGS.1 requirement ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration. The AGD_ADM.1 requirement mandates the developer provide the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the administrator, and how to configure the TOE's access control rulesets for protecting web servers. The documentation also provides a description of how to setup and review the auditing features of the TOE. The AGD_USR.1 is intended for non-administrative users, but could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). Since the non-administrative users of this TOE only interact with the TOE via web agent, it is expected that the user guidance would only discuss the secure access to protected web servers and how the authentication mechanism on the web server is used to pass the user's request for access to web resources to the TOE. AVA_MSU.1 ensures that the guidance documentation is complete, clear, consistent and reasonable. The guidance will define that secure procedures for all modes of operation, including a list of assumptions and requirements for the environment.
O.AUDIT_GENERATION: The TOE will provide the capability to detect and create records of security-relevant events associated with users	FAU_GEN.1-NIAP-0410 FAU_GEN.2-NIAP-0410 FIA_USB.1-NIAP-0351	FAU_GEN.1-NIAP-0410 defines the set of events that the TOE must be capable of recording. This requirement ensures that the Security Administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP. FAU_GEN.2-NIAP-410 ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the "userid". When TOE components imitate actions that need to be audited, the TOE will ensure a mechanism is in place to identity the component as the entity conducting the action. FIA_USB.1 plays a role is satisfying this objective by requiring a binding of security attributes associated with users

Objective	Requirements Addressing the Objective	Rationale
		that are authenticated with the subjects acting on there behalf in the TOE. This only applies to authorized users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail may not always have the proper identity of the subject that causes an audit record to be generated (e.g., invalid userid attempting to gain access to a protected web page or the TOE administration).
O.CONFIGURATION_IDENTIFICATION: The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.	ACM_CAP.2 ALC_FLR.2	ACM_CAP.2 contributes to this objective by requiring the developer provide a reference for the TOE and use a configuration management system CM The developer shall also documentation including a configuration list that describes the configuration items that comprise the TOE. This documentation will describe the method used to uniquely identify the configuration items. ALC_FLR.2 plays a role in satisfying the "analyzed" portion of this objective by requiring the developer to have procedures that address flaws that have been discovered in the product, either through developer actions (e.g., developer testing) or those discovered by others. The flaw remediation process used by the developer corrects any discovered flaws and performs an analysis to ensure new flaws are not created while fixing the discovered flaws.
O.CORRECT_ TSF_OPERATION: The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.	FPT_TST_EXP1.1	O_CORRECT_TSF_OPERATION requires security functional requirements in the FPT class to provide the end user with the capability to ensure the TOE's security mechanisms continue to operate correctly in the field. FPT_TST_EXP1.1.1 is necessary to ensure the correctness of the TSF configuration files and TSF data. FPT_TST_EXP1.1.2 is necessary to ensure the integrity of the TSF executable code. If TSF software is corrupted it is possible that the TSF would no longer be able to enforce the security policies. This also holds true for TSF data, if TSF data is corrupt the TOE may not correctly enforce its security policies. The FPT_TST_EXP1 functional requirement includes the critical nature and specific handling of the cryptographic related TSF data. Since the cryptographic TSF data has specific FIPS PUB requirements associated with them it is important to ensure that any fielded testing on the integrity of these data maintains the same level of scrutiny as specified in the FCS functional requirements.
O.CRYPTOGRAPHY: The TOE shall use NIST FIPS 140-2 validated cryptographic services	FCS_CKM.1 FCS_CKM.2 FCS_CKM.4 FCS_COP.1	The FCS requirements used in this PP satisfy this objective by levying requirements that ensure the cryptographic standards include the NIST FIPS publications (where possible) and NIST approved ANSI standards. The intent is to have the satisfaction of the cryptographic standards be validated through a NIST FIPS 140 validation. FCS_CKM.1specifys the key sizes and standards that are required for the generation of symmetric and asymmetric keys. FCS_CKM.2 is the exception to the other cryptographic requirements in that a NIST approved standard does not exist for key distribution of asymmetric keys. In this case, NIAP-certified DoD PKI for public key distribution using NSA-approved certificate schemes is deemed to be acceptable. FCS_CKM.4 mandates the standards (FIPS 140-2) that must be satisfied when the TOE performs Cryptographic Key Zeroization.

Objective	Requirements Addressing the Objective	Rationale
		FCS_COP.1 requires that for data decryption and encryption that the NIST approved 3DES algorithm be used, and that the algorithm meets the FIPS PUB 140-2, FIPS PUB 46-3, and ANSI X9.52-1998 standards.
O.DISPLAY_BANNER: The TOE will display an advisory warning regarding use of the TOE	FTA_TAB.1	FTA_TAB.1 meets this objective by requiring the TOE display a Security Administrator defined banner before a user can establish an authenticated session. This banner is under complete control of the Security Administrator in which they specify any warnings regarding unauthorized use of the TOE and remove any product or version information if they desire.
O.DOCUMENTED_DESIGN: The design of the TOE is adequately and accurately documented.	ADV_FSP.1 ADV_HLD.1 ADV_RCR.1	There are two different perspectives for this objective. One is from the developer's point of view and the other is from the evaluator's. The ADV class of requirements is levied to aide both parties in the understanding the design documentation necessary for the TOE. ADV_FSP.1 ensures the developer documents the TOE with a functional specification that clearly describes the TSF, including the purpose and method of use of all external TSF interfaces. ADV_HLD.1 requires the developer to provide the high-level design of the TSF. Although the presentation of the high-level design can be informal, it must be internally consistent. The design will also include a description of the security functionality provided by each subsystem of the TSF. Having accurate design documentation is imperative for evaluator's to gain an appropriate level of understanding of the TOE's security operations in a reasonable amount of time. The ADV_RCR.1 is used to ensure that the levels of decomposition of the TOE's design are consistent with one another. This is important, since design decisions that are analyzed and made at one level (e.g., functional specification) that are not correctly designed at a lower level may lead to a design flaw. This requirement helps in the design analysis to ensure design decisions are realized at all levels of the design
O.FAULT_TOLERANCE The TOE will provide limited capabilities to support degraded fault tolerance and fail over for some TOE components.	FRU_FLT.1	FRU_FLT.1 ensures that authorization decision operations can continue to be provided when a single instance of authorization server policy engine fails. An automated fail over mechanism should be provided within the TOE to allow for an alternate authorization server policy engine to make decisions.
O.MANAGE: The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MTD.1 FMT_MSA.1 FMT_MSA.2 FMT_MSA.3-NIAP-409 FMT_MOF.1 FMT_SMR.1	The FMT requirements are used to satisfy this management objective, as well as other objectives that specify the control of functionality. The requirement's rationale for this objective focuses on the administrator's capability to perform management functions in order to control the behavior of security functions The FMT_MTD.1(1) requirement is intended to restrict the ability to change default, query, modify, delete, clear, all TSF data, including system configuration files and cryptographic security data, to the Security Administrator. FMT_MTD.1(2) provides the Security Administrator the

Objective	Requirements Addressing the Objective	Rationale
		capability to manage the TOE's ruleset. This capability is restricted to only the Security Administrator, or another administrative user designated by the Security Administrator, and allows them to create, view, modify and delete the rules that comprise the ruleset. FMT_MSA.1 provides the Security Administrator the capability to manipulate the security attributes associated with both users (stored in the UADS), and protected resources (stored in the Policy Attribute Data Store) that are used for access control permission rules. An example of this would be assigning a specific "group" attribute to several users, stored in the UADS, and then assigning that same "group" attribute to a protected resource in the Policy Attribute Data Store. An access control rule can then be developed to allow access only when the user's group attribute and the resource group attribute match. FMT_MSA.2.1 ensures that only specific secure values are accepted for security attributes. This requirement is designed meet the DCID requirement to prevent user authentication password reuse. A history of static authenticator changes will be maintained with assurance of non-replication of individual authenticators. When a user changing their password submits a previously used password, the system will consider that an "insecure" value for that security attribute and reject it. FMT_MSA.3-NIAP-0409(1) requires that by default, the TOE does not allow an access to a protected resource until a rule in the ruleset allows it. FMT_MOF.1(1) is used by the Security Administrator for user "Account Management". This includes identifying types of accounts (e.g.:individual and group, conditions for group membership, associated privileges, etc). FMT_MOF.1.1(2) is used for the Security Administrator and the Audit Administrator to manage the audit functions of the TOE. This includes modifying the behavior of the Security Audit (FAU_SAR), Security Audit Analysis (FAU_SAA), and the Security Audit (FAU_SEL) functions. This provide those administrators the ability to set or unset specific aspects of the audit mechanism, such as what user behavior is audited FMT_SMR.1 requires that roles exist for administrative actions: the Security Administrator, who is responsible for configuring the TOE's security policies, including the management the security data that is critical to the cryptographic operations; and the Audit Administrator, who is restricted to reading and deleting the audit trail. The TSF is able to associate a human user with one or more roles.
O.MEDIATE: The TOE must protect user data in accordance with its security policy.	FDP_ACC.1 FDP_ACF.1-NIAP-0460	The FDP requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation takes place on an authorization server. FDP_ACC.1 defines that an Authorization Server Access Control policy will be enforced on a list of subjects acting on the behalf of users attempting to gain access to a list of named objects. All the operations among subject and object covered are by the Authorization Server policy. The "subjects" are generally the Authorization Server "Agents." The "named objects" are the designated web based resources (web server, directories, files, or objects) that the Authorization Server is protecting. FDP_ACF.1.-NIAP-0460 defines the Security Attribute used to provide Access Control to objects based on the following Authorization Server Access Control policy

Objective	Requirements Addressing the Objective	Rationale
O.PARTIAL_FUNCTIONAL_ TESTING The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	ATE_COV.1 ATE_FUN.1 ATE_IND.2	In order to satisfy O.PARTIAL_FUNCTIONAL_ TESTING, the ATE class of requirements is necessary. The component ATE_FUN.1 requires the developer to provide the necessary test documentation to allow for an independent analysis of the developer's security functional test coverage. ATE_COV.1 requires the developer to provide a test coverage analysis that demonstrates the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification. While exhaustive testing of the TSFI is not required, this component ensures that the security functionality of each TSFI is addressed. This component also requires an independent confirmation of the completeness of the test suite, which aids in ensuring that correct security relevant functionality of a TSFI is demonstrated through the testing effort. ATE_IND.2 requires an independent confirmation of the developer's test results, by mandating a subset of the test suite be run by an independent party. Upon successful adherence to these requirements, the TOE's conformance to the specified security functional requirements will have been demonstrated.
O.PARTIAL_SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.	FPT_SEP_EXP.1 FPT_SEP_EXP.2 FPT_RVM.1 FPT_FLS.1 ADV_SPM.1 FPT_ITC.1 FPT_ITT.1 FPT_RCV.NIAP-0389-1	FPT_SEP was chosen to ensure the TSF provides a domain that protects itself from untrusted users. If the TSF cannot protect itself it cannot be relied upon to enforce its security policies. Since "Software Only" technology cannot fully meet the FPT_SEP requirements as written, Software Only TOEs are expected to work in the context of their hardware environment to aid in enforcing domain separation. Therefore, the inclusion of the *explicitly stated* requirements FPT_SEP_EXP.1 and FPT_SEP_EXP.2 are used to define the domain separation between the security domains of subjects in the TOE Scope of Control. The inclusion of FPT_RVM.1 ensures that the TSF makes policy decisions on all interfaces that perform operations on subjects and objects that are scoped by the policies. Without this non-bypassability requirement, the TSF could not be relied upon to completely enforce the security policies, since an interface(s) may otherwise exist that would provide a user with access to TOE resources (including TSF data and executable code) regardless of the defined policies. This includes controlling the accessibility to interfaces, as well as what access control is provided within the interfaces. The inclusion of FPT_FLS.1 and AVM_SPM.1 provide dependency support to the degraded fault tolerance requirement FRU_FLT.1. When a TOE software component fails, FPT_FLS.1 ensures it fails to a secure state. The ACM_SPM.1 ensures there is an informal security policy that addresses the protection of the TOE when it fails to a secure state. FTP_ITC.1 and FTP_ITT are necessary to protect TSF data during communication between TOE components and for communications with other trusted IT entities (e.g., user and policy data stores) from unauthorized disclosures. These requirements also ensure a secure path between the TOE and remote administrators The protection of the communication path when TSF data is being transmitted is critical to the TSF maintaining a domain of execution that cannot be tampered or interfered with, thus resulting is a possible unauthorized disclosure or security policy failure. FPT_RCV.NIAP-0389-1 ensures that the TOE does not continue to operate in an insecure state when software failure occurs. Upon the failure a TOE software component, the TOE will automatically enter a state that disallows access control

Objective	Requirements Addressing the Objective	Rationale
		decisions and requires an administrator to follow documented procedures that instruct them on to return the TOE to a secure state. These procedures may include running diagnostics of the software or utilities that may correct any integrity problems found with the TSF data or code. It may even require that the administrator reload and install the TOE software from scratch.
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.	FDP_RIP.2 FCS_CKM_EXP.2 FCS_CKM.4	FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data. For this TOE it is critical that the memory used to make authorization decisions is either cleared or that some buffer management scheme be employed to prevent the authorization decision of one user's request to be used in a subsequent authorization decision. FCS_CKM_EXP.2 places requirements on how cryptographic keys are managed within the TOE. This requirement places restrictions in addition to FDP_RIP.2, in that when a cryptographic key is moved from one location to another (e.g., calculated in some scratch memory and moved to a permanent location) that the memory area is immediately cleared as opposed to waiting until the memory is reallocated to another subject. FCS_CKM.4 applies to the destruction of cryptographic keys used by the TSF. This requirement specifies how and when cryptographic keys must be destroyed. The proper destruction of these keys is critical in ensuring the content of these keys cannot possibly be disclosed when a resource is reallocated to a user.
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_AFL.1-NIAP-0425 FIA_ATD.1 FIA_SOS.1 FIA_UID.1 FIA_UAU.1 FIA_UAU.5 AVA_SOF.1 FTA_MCS.2 FTA_SSL.1 FTA_SSL.2 FTA_SSL.3	FIA_AFL.1-NIAP-0425 provides a detection mechanism for unsuccessful authentication attempts by remote administrators. The requirement enables a Security Administrator settable threshold that prevents unauthorized users from gaining access to authorized administrator's account by guessing authentication data by locking the targeted account until the Security Administrator takes some action (e.g., re-enables the account) or for some Security Administrator defined time period. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE. FIA_ATD.1 defines the attributes for administrators and users, including a userid that is used to by the TOE to determine a user's identity and enforce what type of access the user has to the TOE (e.g., the TOE associates a userid with any role(s) they may assume) or to another protected resource based on the access control policy. This requirement allows a human user to have more than one user identity assigned, so that a single human user could assume all the roles necessary to manage the TOE. In order to ensure a separation of roles, this PP requires a single role to be associated with a user id. This is inconvenient in that the administrator would be required to log in with a different user id each time they wish to assume a different role, but this helps mitigate the risk that could occur if an administrator were to execute malicious code. FIA_SOS.1.1 ensures that a mechanism is in place to verify that user's passwords must contain a minimum of 8 alphanumeric charters with at least one numeric charter. This type of password cannot be easily be broken with a dictionary search or elementary password cracking software. FIA_UAU.1 contributes to this objective by limiting the

Objective	Requirements Addressing the Objective	Rationale
		services that are provided by the TOE to unauthenticated users. Verification of number of concurrent sessions established and auditing of attempted session establishment on behalf of the administrative user are the only performed before the administrative user is authenticated. FIA_UAU_5 ensures the TOE can allow access to a protected resource via multiple authentication mechanisms. At a minimum the TOE shall support user ID and password, X.509V3 identity certificates in software, and X.509V3 identity certificates from hardware tokens. This requirement also contributes to the objective by allowing Group authenticators to be used in conjunction with the use of an individual/unique authenticator. The AVA_SOF.1 requirement is applied to the administrator authentication mechanism. For this TOE, the strength of function specified is basic. This requirement ensures the developer has performed an analysis of the authentication mechanism to ensure the probability of guessing a user's authentication data would require a medium-attack potential. FTA_MCS.2 contributes to the controlling access to the TOE by restricting the maximum number of concurrent sessions that belong to the same administrative user according to the rules set by the security administrator. The system will enforce a default number on sessions determined by security administrator. Restricting the number of concurrent sessions reduces the threat unauthorized access by personnel which have obtained an authorized user's identity credentials and are logging in from a deferent location from the authorized user (given one user should not be in two places at the same time). The FTA_SSL family partially satisfies the O.TOE_ACCESS objective by ensuring that user's sessions are afforded some level of protection. FTA_SSL.1 provides the Security Administrator the capability to specify a time interval of inactivity in which an unattended administrative session would be locked and will require the administrator responsible for that session to re-authenticate before the session can be used to access TOE resources. FTA_SSL.2 provides administrators the ability to lock their administrative session. This component allows administrators to protect their session immediately, rather than waiting for the time-out period and minimizes their session's risk of exposure. FTA_SSL.3 allows the TSF to terminate administrative sessions after a Security Administrator defined time interval of inactivity.
O.VULNERABILITY_ ANALYSIS The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	AVA_VLA.1	To maintain consistency with the overall assurance goals of this TOE, O.VULNERABILITY_ANALYSIS requires the AVA_VLA.1 component to provide the basic level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated. AVA_VLA.1 requires the developer to perform a search for obvious ways in which a user can violate the TSP. The developer will document the disposition of obvious vulnerabilities. For those vulnerabilities that are not eliminated the developer will show that the vulnerability cannot be exploited in the intended environment for the TOE. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent of low attack potential to violate the TOE's security policies.

6.4 Rationale for Assurance Requirements

EAL2 augmented was chosen to ensure an adequate level of confidence in security services used to protect information in Basic Robustness Environments. The assurance selection was based on the postulated low threat environment.

The EAL definitions in Part 3 of the CC were reviewed and the Basic Robustness Assurance Package (EAL2 augmented with assurance requirements ALC_FLR.2, and AVA_MSU.1) was believed to best achieve this goal. The sponsor concluded that EAL2 augmented is applicable since this PP addresses circumstances where developers and users require a basic to moderate level of independently assured security in commercial products. The addition of assurance requirement AVA_VLA.1 ensures that the developer vulnerability analysis is done to demonstrate the resistance to penetration attackers with low attack potential and that a systematic approach is used to search for obvious vulnerabilities. This collection of assurance requirements require TOE developers to gain assurance from good software engineering development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. Rationale for individual assurance requirements is provided in Table 13.

The postulated threat environment specified in Section 3 of this PP was used in conjunction with the Information Assurance Technical Framework (IATF) Robustness Strategy guidance to derive the chosen assurance level.

These factors were taken into consideration and the conclusion was that the basic robustness assurance package was the appropriate level of assurance.

6.5 Rationale for Strength of Function Claim

Part 1 of the CC defines “strength of function” in terms of the minimum efforts assumed necessary to defeat the expected security behavior of a TOE security function. There are three strength of function levels defined in Part 1: SOF-basic, SOF-medium and SOF-high. SOF-basic is the strength of function level chosen for this PP TOE is SOF-basic. The evaluated TOE is intended to operate in DoD basic robustness environments processing classified information. Users and administrator in a DoD environment will have a clearance to access all data processed by the TOE, but not necessarily the need to know. All users are assumed to be cooperative and non-malicious. In commercial environments, company sensitive information may be processed, with users being cooperative, and not likely to attempt sophisticated attacks at data for which they are not authorized.

6.6 Rational for Satisfying all Dependencies

The Intrusion Authorization Server Protection Profile does satisfy all the requirement dependencies of the Common Criteria. Table 14 lists each requirement from the Authorization Server Protection Profile with a dependency and indicates whether the dependent requirement was included. As the table indicates, all dependencies have been met.

Table 14 – Requirement Dependencies

Functional Component	Dependencies	Included
FAU_GEN.1.2-NIAP-0460	FPT_STM.1	Yes, in IT Environment
FAU_SAR.1	FAU_GEN.1	Yes
FAU_SAR.2	FAU_SAR.1	Yes
FAU_SAR.3	FAU_SAR.1	Yes
FAU_SEL.1-NIAP-0460	FAU_GEN.1, FMT_MTD.1	Yes
FAU_STG.1-NIAP-0460	FAU_GEN.1	Yes
FAU_STG.3	FAU_STG.1	Yes
FCS_CKM.1	FCS_CKM.2	Yes
FCS_CKM.2	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	Yes
FCS_CKM.4	FCS_CKM.1, FMT_MSA.2	Yes
FCS_COP.1	FCS_CKM.1,FCS_CKM.4, FMT_MSA.2	Yes
FDP_ACC.1	FDP_ACF.1	Yes
FDP_ACF.1-NIAP-0460	FDP_ACC.1, FMT_MSA.3	Yes
FIA_USB.1	FIA_ATD.1	Yes
FMT_MOF.1	FMT_SMR.1	Yes
FMT_MSA.1	FDP_ACC.1	Yes
FMT_MSA.2	ADV_SPM.1, FDP_ACC.1	Yes
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	Yes
FMT_MTD.1	FMT_MSA.1,FMT_SMR.1	Yes
FMT_SMR.1	FIA_UID.1	Yes
FPT_RCV.NIAP-0389-1	FPT_TST.1, AGD_ADM.1, ADV_SPM.1	Yes
FRU_FLT.1	FPT_FLS.1	Yes
FTA_MCS.2	FIA_UID.1	Yes
FTA_SSL.1	FIA_UAU.1	Yes
FTA_SSL.2	FIA_UAU.1	Yes

6.7 Rationale for Explicit requirements

Table 15 presents the rationale for the inclusion of the explicit requirements found in this PP.

Table 15 – Rational for Explicit Requirements

Explicit Requirement	Identifier	Rationale
FPT_SEP_EXP.1	TSF Domain Separation	This explicit requirement is necessary since the CC does not specifically provide for the Domain Separation requirements for software only PPs

Explicit Requirement	Identifier	Rationale
FPT_SEP_EXP.2	TSF Domain Separation	This explicit requirement is necessary since the CC does not specifically provide for the Domain Separation requirements for software only PPs
FPT_TST.EXP1.1	TSF Testing	This explicit requirement is necessary since the CC does not specifically provide for “software only” PP. Following the “basic guidance” reduces the potential inconsistencies amongst Basic Robustness TOE

7 REFERENCES

1) Common Criteria for Information Technology Security Evaluation*, CCIB-98-031 Version 2.1, August 1999.*

2) Information Assurance Technical Framework, *Version 3.0, September 2000.*

3) Federal Information Processing Standard Publication (FIPS-PUB) 46-3, Data Encryption Standard (DES), October 1999.

4) Federal Information Processing Standard Publication (FIPS-PUB) 140-2, Security Requirements for Cryptographic Modules, May 25, 2001.

5) Internet Engineering Task Force, IP Encapsulating Security Payload (ESP), RFC 2406, November 1998.

6) Internet Engineering Task Force, Internet Key Exchange (IKE), RFC 2409, November 1998.

7) Internet Engineering Task Force, ESP CBC-Mode Cipher Algorithms, RFC 2451, November 1998.

8) Internet Engineering Task Force, Use of HMAC-SHA-1-96 within ESP and AH, RFC 2404, November 1998.

9) Department of Defense Instruction, Information Assurance Implementation Draft *No. 8500.bb, September 2001*.

10) The AES Cipher Algorithm and Its Use with IPsec <draft-ietf-ipsec-ciph-aes-cbc.03.txt>, *Internet draft, November 2001.*

11) Federal Information Processing Standard Publication (FIPS-PUB) 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001

8 TERMINOLOGY

In the Common Criteria, many terms are defined in Section 2.3 of Part 1. The following are a definitions of terms used in this PP and common to other DoD PPs.

Access -- Interaction between an entity and an object that results in the flow or modification of data.

Access Control -- Security service that controls the use of resources[1] and the disclosure and modification of data.[2]

Accountability -- Property that allows activities in an IT system to be traced to the entity responsible for the activity.

Administrator -- A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Assurance -- A measure of confidence that the security features of an IT system are sufficient to enforce its' security policy.

Asymmetric Cryptographic System -- A system involving two related transformations; one determined by a public key (the public transformation), and another determined by a private key (the private transformation) with the property that it is computationally infeasible to determine the private transformation (or the private key) from knowledge of the public transformation (and the public key).

Asymmetric Key -- The corresponding public/private key pair needed to determine the behavior of the public/private transformations that comprise an asymmetric cryptographic system.

Attack -- An intentional act attempting to violate the security policy of an IT system.

Authentication -- Security measure that verifies a claimed identity.

Authentication data -- Information used to verify a claimed identity.

---

[1] Hardware and software.

[2] Stored or communicated.

Authorization -- Permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized user -- An authenticated user who may, in accordance with the TSP, perform an operation.

Availability -- Timely[3], reliable access to IT resources.

Compromise -- Violation of a security policy.

Confidentiality -- A security policy pertaining to disclosure of data.

Critical Security Parameters (CSP) -- Security-related information (e.g., cryptographic keys, authentication data such as passwords and pins, and cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

Cryptographic Administrator -- An authorized user who has been granted the authority to perform cryptographic initialization and management functions. These users are expected to use this authority only in the manner prescribed by the guidance given to them.

Cryptographic boundary -- An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.

Cryptographic key (key) -- A parameter used in conjunction with a cryptographic algorithm that determines [7]:

- the transformation of plaintext data into ciphertext data,
- the transformation of cipher text data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a data authentication code computed from data.

Cryptographic Module -- The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module.

Cryptographic Module Security Policy -- A precise specification of the security rules under which a cryptographic module must operate, including the rules derived from the requirements of this PP and additional rules imposed by the vendor.

---

[3] According to a defined metric.

Defense-in-Depth (DID) -- A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.

Discretionary Access Control (DAC) -- A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. These controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

DMZ -- A Demilitarized Zone (DMZ) is a network that is mediated by the TOE but, as a result of less stringent access controls, provides access to publicly available services, such as web servers.

Embedded Cryptographic Module -- One that is built as an integral part of a larger and more general surrounding system (i.e., one that is not easily removable from the surrounding system).

Enclave -- A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or may be based on physical location and proximity.

Entity -- A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

External IT entity -- Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

Identity -- A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity -- A security policy pertaining to the corruption of data and TSF mechanisms.

Integrity label -- A security attribute that represents the integrity level of a subject or an object. Integrity labels are used by the TOE as the basis for mandatory integrity control decisions.

Integrity level -- The combination of a hierarchical level and an optional set of non-hierarchical categories that represent the integrity of data.

Mandatory Access Control (MAC) -- A means of restricting access to objects based on subject and object sensitivity labels.[4]

Mandatory Integrity Control (MIC) -- A means of restricting access to objects based on subject and object integrity labels.

---

[4] The Bell LaPadula model is an example of Mandatory Access Control

Multilevel -- The ability to simultaneously handle (e.g., share, process) multiple levels of data, while allowing users at different sensitivity levels to access the system concurrently. The system permits each user to access only the data to which they are authorized access.

Named Object[5] -- An object that exhibits all of the following characteristics:

- The object may be used to transfer information between subjects of differing user identities within the TSF.
- Subjects in the TOE must be able to request a specific instance of the object.
- The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

(Note: Due to the deletion of the last sentence in the OS PP (pertaining to intended use of the object being for sharing user data), something may need to be done to the requirements section of the PP (i.e., FDP_ACF) to ensure that some objects, which may satisfy the above but which are not intended for sharing user data do not need a full DAC implementation but rather it is acceptable if they are "owner only" or some other appropriate mechanism.)

Non-Repudiation -- A security policy pertaining to providing one or more of the following:

- To the sender of data, proof of delivery to the intended recipient,
- To the recipient of data, proof of the identity of the user who sent the data.

Object -- An entity within the TSC that contains or receives information and upon which subjects perform operations.

Operating Environment -- The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

Operating System (OS) -- An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.

Operational key -- Key intended for protection of operational information or for the production or secure electrical transmissions of key streams.

Peer TOEs -- Mutually authenticated TOEs that interact to enforce a common security policy.

---

[5]The only named objects in this PP, are operating system controlled files.

Public Object -- An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.

Robustness -- A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly. DoD has three levels of robustness:

- **Basic:** Security services and mechanisms that equate to good commercial practices. Basic robustness equates to EAL-2 plus; ALC_FLR (Flaw Remediation), and AVA_MSU.1 (Misuse-Examination Guidance) as defined in CCIB-98-028, Part 3, Version 2.0

- **Medium:** Security services and mechanisms that provide for layering of additional safeguards above good commercial practices. Medium robustness equates to EAL-4 plus; ADV_IMP.2, ADV_INT.1, ALC_FLR.2, ATE_DPT.2, and AVA_VLA.3 (Moderately Resistant Vulnerability Analysis) as defined in CCIB-98-028, Part 3, Version 2.0. If cryptographic functions are included in the TOE, then AVA_CCA_EXP.2 is also included as documented in the Protection Profile Medium Robustness Consistency Guidance.

- **High:** Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.

Secure State -- Condition in which all TOE security policies are enforced.

Security attributes -- TSF data associated with subjects, objects, and users that is used for the enforcement of the TSP.

Security level -- The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity on the information [10].

Sensitivity label -- A security attribute that represents the security level of an object and that describes the sensitivity (e.g.,Classification) of the data in the object. Sensitivity labels are used by the TOE as the basis for mandatory access control decisions [10].

Split key -- A variable that consists of two or more components that must be combined to form the operational key variable. The combining process excludes concatenation or interleaving of component variables.

Subject -- An entity within the TSC that causes operations to be performed.

Symmetric key -- A single, secret key used for both encryption and decryption in symmetric cryptographic algorithms.

Threat -- Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

Threat Agent - Any human user or Information Technology (IT) product or system which may attempt to violate the TSP and perform an unauthorized operation with the TOE.

User -- Any entity (human user or external IT entity) outside the TOE that interacts with the TOE. In the case of the Authorization Server protecting web resources, an "agent" acts on behalf of a user. Therefore, the "user" never truly interacts with the TOE. The authorization server software must have access to the "user's" privilege attributes which are generally maintained in a separate data storage (not part of the TOE).

Vulnerability -- A weakness that can be exploited to violate the TOE security policy.

9 ACRONYMS

The following abbreviations from the Common Criteria are used in this Protection Profile:

AES Advanced Encryption Standard

ATM Asynchronous Transfer Method

CC Common Criteria for Information Technology Security Evaluation

DES Data Encryption Standard

DoD Department of Defense

DMZ Demilitarized zone

EAL Evaluation Assurance Level

ESP Encapsulating Security Payload

FIPS PUB Federal Information Processing Standard Publication

FTP File Transfer Protocol

GIG Global Information Grid

HTTP Hypertext Transfer Protocol

I&A Identification and Authentication

IATF Information Assurance Technical Framework

ICMP Internet Control Message Protocol

IETF Internet Engineering Task Force

IKE Internet Key Exchange

IPSEC ESP Internet Protocol Security Encapsulating Security Payload

IP Internet Protocol

IT Information Technology

MRE Medium Robustness Environment

NBIAT&S Network Boundary Information Assurance Technologies and Solutions Support

NIAP National Information Assurance Partnership

NIST National Institute of Standards and Technology

NSA National Security Agency

NTP Network Time Protocol

PKI Public Key Infrastructure

PP Protection Profile

RNG Random Number Generator

SFP Security Function Policy

SMTP Simple Mail Transfer Protocol

SOF Strength of Function

ST Security Target

TCP Transmission Control Protocol

TFTP Trivial File Transfer Protocol

TOE Target of Evaluation

TSE TOE Security Environment

TSF TOE Security Function

TSP TOE Security Policy

UDP User Datagram Protocol

URL Uniform Research Locator

VPN Virtual Private Network